1-1-1970

A data flow controller and refresh memory for a computer display system.

David Logan Murray
University of Windsor

A DATA FLOW CONTROLLER AND REFRESH MEMORY FOR A COMPUTER DISPLAY SYSTEM

by

DAVID LOGAN MURRAY

A Thesis

Submitted to the Faculty of Graduate Studies through the Department of Electrical Engineering in Partial Fulfillment of the Requirement for the Degree of Master of Applied Science at the University of Windsor

Windsor, Ontario
1970

APPROVED:

321823

ABSTRACT

This thesis presents the design and implementation of a digital controller and high speed refresh memory for a cathode ray tube (CRT) display terminal.

The specific function of this device was the control of and the supply of data to graphical and alphanumeric display generators that were also in the development stages. The system for CRT display was to be used in conjunction with a Digital Equipment Corporation (DEC) PDP8/S computer.

The slow cycle speed and single stream processing of the PDP8/S made the development of control and refresh circuitry necessary to free the computing facilities of the PDP8/S for programmed data processing necessary for the effective use of CRT display equipment.

The device described in this thesis was constructed and interfaced to the PDP8/S through the standard input-output (I/O) facilities of the PDP8/S which "sees" the device as one of its peripherals. Thus the PDP8/S has been fitted with CRT display facilities over which it exerts indirect control through the logic presented in this thesis. The PDP8/S is, therefore, not operating as a dedicated computer but is free to perform any other program tasks and I/O to its other peripherals while maintaining its CRT display in full operation.

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to Dr. P.A.V. Thomas who supervised this work and for his helpful suggestions. Acknowledgement must also go to the National Research Council of Canada for the financial aid which made this project possible.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER I

INTRODUCTION

1.1 General Aspects and Classification of the Problem.

The display of visual information may be divided into three areas; preparation of data into the appropriate digital coding, control of data flow and rate of flow, and the processing of this data into a form suitable for display. This thesis deals with a device to control data flow and rate of flow in a particular CRT display system.

In general, visual information may be separated into two classifications; text and graphical. Text generally consists of alphanumeric information but may also include any special characters commonly used. Graphical data consists of information necessary for the display of drawings, figures, graphs, or any non-standard shapes. Graphical data may be subdivided into further categories such as vector (straight line), dot, circle, sector, etc. The classification of visual display data is subject to interpretation. However, the importance of data classification to the control of data flow lies in the fact that different types of information must be processed for CRT display by different Display Generators. Thus, information must be channelled to the appropriate generator regardless of the hardware,

software, or firmware nature of those generators. The device which oversees the data flow must direct the data into the proper channel and in the proper sequence. This device or controller may be software, or hardware in nature or a combination of both.

A second problem involved in the CRT display of visual data is that of rate of flow. Two types of CRT display are possible; the first employs a memory tube which retains any trace that is applied to it, and the second uses a conventional CRT which requires a constant and rapid retrace or refresh of a trace and depends on the image retention of the human eye to hold a constant image. The former approach is the simpler to implement but its use removes the possibility of man-machine interaction through the use of a light pen (Chapter II); the latter method although more costly and difficult to implement facilitates the use of the light pen and it was the approach used in the device described in this thesis. Discussion will, therefore, be limited to the rapid refresh approach.

The refresh rate of information displayed on a CRT depends upon the phosphors used in the CRT itself but a lower limit of this refresh rate can generally be set at thirty frames per second necessary for a non-flickering image retention with phosphors commonly used. Thus, visual data not only has to be channelled to the appropriate generator but also at the correct rate. That is, fast enough to maintain a reasonable amount of data displayed on the CRT at a

refresh rate adequate to insure non-flickering image retention but not faster than the generators can process the data for display. The problem is further complicated by the fact that the different generators will likely process their data at different rates.

The remainder of this thesis deals with a particular data flow problem and its solution.

1.2 The Problem and Design Criteria.

Based on research previously carried out in the Electrical Engineering Department of the University of Windsor [1][2] and on research currently in progress [3] interest became aroused in the construction of a CRT display terminal. After careful consideration the design problem involved in developing a CRT display system divided itself into three areas: the design and implementation of an alphanumeric (A/N) character generator, the design and implementation of a vector and dot generator, and the design and implementation of a data flow controller and refresh circuit. The latter design area is dealt with in this thesis.

Both the control of data flow and the refresh of a CRT may be implemented by software techniques. The only computer available for this project was the DEC PDP8/S.[4] A quick examination of the feasibility of using the PDP8/S for a software implementation of control and refresh led to the following minimum cycle time for the output of a single piece of data:

Machine Language Program Step		Execution Time in Microseconds
Clear Accumulator		28
Two's Compliment Add		36
I/O Instruction		38
	Total	102 μsec

The above program sequence represents the absolute minimum time required to output a single word of data and in practice the average output time would be much higher due to program decisions regarding data channelling and necessary program looping. However, even if we consider the above 102 μsec as the average cycle time for a single piece of data and if we take the lower limit of thirty frames per second refresh rate the maximum number of data words that may be displayed on a CRT in one image is:

$$(1/30 \text{ sec})/(102 \times 10^{-6} \text{ sec}) = 328$$

or approximately five of the printed lines on this page if we consider A/N data only. Thus, if the PDP8/S was to handle the refresh of the CRT display the amount of data available for display would be seriously limited. Perhaps an even more restricting factor lies in the fact that the programmed data processing capabilities of the PDP8/S would be almost completely tied up with refresh and control of data flow and would not be available for programmed data manipulation that makes a CRT display terminal a useful device. It, therefore, became obvious that some type of a hardware circuit would be necessary to handle the data flow control and refresh of the CRT to achieve a higher rate of

data flow and to remove the burden of these routine operations from the processor of the PDP8/S.

After the decision was reached to design and implement a hardware control and refresh device it was necessary to decide upon a set of necessary and desirable operating features. The list of operating features deemed necessary for minimum acceptable operation were as follows:

(i) This device must have some type of memory capable of storing a display file and cycling through this display file to supply data to display generators at a rate as fast as the generators could process the data for CRT display. This would insure the display of the maximum amount of data at an acceptable refresh rate.

(ii) This device must have the ability to accept data from the I/O channel of the PDP8/S for the purpose of building up the contents of its display file. Thus, the contents of the display file would be under the control of the PDP8/S while the refresh of this data would be handled by the external refresh memory.

(iii) Since data of several types (i.e. A/N, dot, vector) was to be displayed, this device must have the ability to channel data to the appropriate generator.

The above list of design criteria were necessary to the minimum acceptable operation. However, it was recognized that

certain other features would greatly enhance the overall flexibility and usefulness of the system. These additional features are listed below:

(i) The ability of this device to hold in its memory multiple display files of variable length coupled with the ability to select one of these for refresh under PDP8/S control would greatly improve the systems smoothness of operation while reducing the storage requirements of the memory of the PDP8/S.

(ii) The ability of this device to accept single or block data changes, insertions, deletions or additions to any of its display files whether or not they are currently being refreshed without interrupting the display would make possible very effective on-line editing of information.

(iii) In the event that man-machine interaction would be implemented the potential for this device to retain the position of any piece of data in the display file upon a light pen interrupt would be essential for the possible processing of this particular data word by the PDP8/S.

One further restraint was placed on the design of the control and refresh circuits. Since the design of the display generators was to proceed simultaneously with the design of the data flow controller and refresh circuitry, the exact nature and characteristics of the generators was not available. In addition, it was probable that

additional display generators would be added to the system after completion of the data flow and refresh logic. Therefore, it was necessary to design this device to supply data to an undetermined number of generators that may operate with a wide variety of timing requirements.

CHAPTER II

DESIGN PHILOSOPHY

2.1 Preliminary Design Decisions.

Before the design could proceed in any detail a decision had to be made on the type of memory to be used for the storage of the display file and for the refresh of the CRT. Three types of memories were readily available on the market for possible use in this application: magnetic tape, magnetic rotating memories (i.e. disk, drum or delay line)and magnetic core memories.[5] Magnetic tape memories were immediately eliminated due to the excessive rewind time that would make refresh cycling impractical. The choice, then, had to be made between a rotating type of memory and a random access core memory. A consideration that had to be kept in mind during the choosing of a specific memory was that of electronic compatibility to the existing equipment. Both the PDP8/S and the A/N generator, which was already in the construction stage, used the 0 and -3 volt logic levels of DEC standard negative logic modules. Thus, any memory used along with the controller circuitry would have to be compatible with these logic levels or logic level conversion would have to be built in at any interface.

Rotating memories seemed to have some natural advantages

over random access core memories because of their circulating nature. At first glance it appeared that refresh of data could be easily implemented using a rotating memory. Moreover, rotating memories tend to be less expensive per bit of storage capacity than random access core memories. However, since rotating memories circulate at a fixed rate any device accepting data from a rotating memory must do so at a rate dictated by the rate of rotation of the memory. Data may be accessed a word at a time in some rotating memories but this approach of random addressing leads to greatly increased access times that would not permit a reasonable refresh rate. This problem may be alleviated to a certain extent by the interlacing of data on the rotating memory tracks. At best, however, the use of a rotating memory requires that the display generators accept data at some fixed rate which puts heavy restrictions on the efficient design of these generators. Rotating memories by their very nature are best suited for block data transfers. Thus, if effective editing of the display file was to take place a very efficient system of memory mapping and data allocation would have to be maintained in the software of the PDP8/S to insure that the refresh rate would not fall below an acceptable minimum due to excessive access time caused by excessive memory address branching.

On the other hand, the random access core memory had the disadvantage that it had to be equipped with a memory address decod-

ing circuit.[5] This was not a particularly serious handicap, however, since a rotating memory would have to be equipped with addressing logic of comparable complexity if it was to be any use at all. The random access core memory does not suffer the disadvantages of a fixed cycling frequency and outputs a data word from any specified location on command. Therefore, if a random access core memory were used the generators receiving data from it could set their own pace (as long as the cycle time was not lower than the minimum cycle time of the memory) and so remove any synchronization problems. In addition, since this type of memory may access any address in the same minimum cycle time, extensive looping and branching could be implemented without careful consideration of memory space allocations and without increasing the average word access time.

It appeared that the random access core memory had operating characteristics superior to those of rotating memories for this application. The last consideration was one of cost. The logical choice of a random access core memory for investigation was the DEC H201 memory stack [6] as it was compatible with all the existing equipment in both logic levels and word length. This memory has a 12 bit data word length, 4096 words of storage, and a minimum cycle time of 8 μ sec. Decoding logic was also available from DEC. The cost of the memory plus decoding and driving logic was approximately $3,300.00. Two rotating memories were also investigated, the Information Data

Systems (IDS) series 8000 disk memory and the DEC disk model DS32. The IDS series 8000 disk has a storage capacity of 5,800 12-bit words with a word transfer rate of 23 μsec per word. This disk memory uses a preamble and postamble approach[7] to data block addressing, thus, random access of data words is extremely wasteful in both memory storage space and in access time with this memory type. The basic cost of the IDS disk was approximately $2,700.00 not including timing and addressing logic and logic level conversion necessary with use of this memory. The DEC disk model DS32 has a storage capacity of 32,000 12-bit words and a word transfer rate of 66 μsec. This disk is equipped for random access of data words with an average access time of 16.67 μsec. The price quoted from DEC for the model DS32 was $3,270.00 and again this price does not include timing and addressing logic.

From the design criteria and cost considerations it was obvious that the DEC H201 random access core memory would provide much superior operating characteristics at a slight penalty in price and a more serious penalty in storage capacity. However, since this memory was primarily for use as a rapid refresh memory and would not be used for bulk storage, it was decided that the random access core memory would provide by far the best compromise. Hence, an H201 memory stack along with its suggested decoding and driving logic was used.

2.2 System Design Concepts.

2.2.1 Introduction

After the choice of a memory type, it became necessary to reach certain design concepts that would make it possible to satisfy as many of the design criteria as possible. The system conceptualization described in the remainder of this chapter demonstrated the ability to satisfy all of the design criteria. For the sake of continuity this description does not follow in the chronological order that these concepts were reached. Instead, the system concepts are presented in an order which will make the overall picture clear. The block diagram and summary at the end of this section will attempt to tie each of the concepts presented into a unified picture of the system and to point out how each of the design criteria outlined in Chapter I is satisfied.

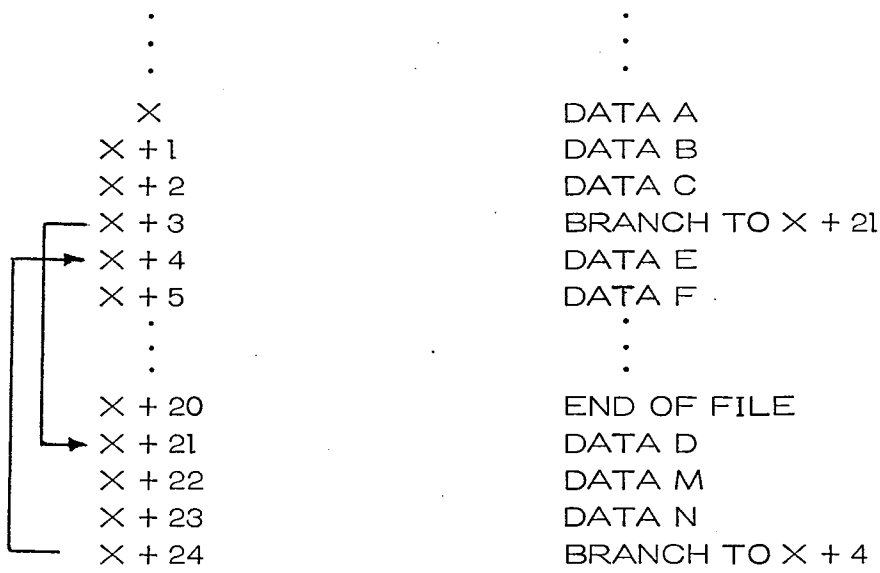
2.2.2 Refresh Addressing, Looping and Branching by Imbedded Instructions.

One of the most attractive features of the random access core memory was its ability to access data words from any location in its storage without an increase in access time. It was decided to exploit this ability as much as possible to improve the flexibility of the system. Editing of a display file (i.e. changes, deletions, or insertions) can most easily be achieved by introducing branch instructions into memory addressing. For example, the

following display file may be considered typical:

Memory Location	Contents
⋮	⋮
X	DATA A
X + 1	DATA B
X + 2	DATA C
X + 3	DATA D
X + 4	DATA E
X + 5	DATA F
⋮	⋮
X + 20	END OF FILE

If it becomes desirable to insert a block of data (of any length) between DATA C and DATA D, the following modified display file employing "branch instructions" embedded in it would achieve this very nicely.



Thus, DATA M and DATA N have been inserted into the display file with a minimum of display file rewriting and with the addition of only two non-data memory cycles. It becomes obvious that a circulating or looping ability can be implemented by placing a "branch to begin-

ning of file instruction" as the last word in the data file. In this way the display file will circulate to provide a refresh for the CRT.

It was evident that an addressing system should be developed that would increment by one the memory address for each word of "data" that was read out and would modify this address to any memory location on command of an imbedded branch or jump instruction. In practice a 12-bit address is required to access all of the 4096 memory locations in the H201 memory stack. Since the basic word length of this memory is only twelve bits, more than one instruction word would be necessary if this instruction was to be recognizable as an instruction, and capable of modifying a 12-bit address. The decision was made to divide the memory into a "page" type of system and use two types of jump instructions; direct jumps and indirect jumps. For direct jumps or one word jumps, the memory was divided into 16 pages each of 256 memory locations. The direct jump instruction modifies the last eight bits of the memory address while leaving the first four bits as they were so that any of the 256 memory locations may be addressed on the current page which is defined by the first four bits of the memory address. Thus, the first four bits of the direct jump instruction are used for instruction identification, while the last eight are used to modify the memory address. One bit of the first four in the direct jump instruction word was set aside for a jump to page "0" function. This allowed the option to modify the

memory address to address any location on the current page or the same location relative to page "0" (i.e. first four bits of memory address are 0000). Thus, the direct jump instruction provides a one word branch instruction that can modify the memory address to access any of the 256 memory locations on the current page or any of the memory locations from 0 to 256 (i.e. page '0'). The indirect jump instruction allows an absolute branch to any location in memory. This function requires two consecutive words in memory. The first word is recognized as an indirect jump instruction and initiates a read of the next consecutive word in memory the twelve bits of which are used as the new memory address. Thus, a full 12-bit address has been selected for the next memory read.

The above addressing system requires the use of two 12-bit registers; a display address register (DAR) which must have the ability to act as a ones counter and the ability to be modified for branching, and a display buffer register (DBR) which receives data from the memory outputs and acts as a decoding register for embedded instructions as well as a buffer register for data words destined for display generators and the contents of which may be used to modify the DAR. Figure I gives a graphical representation of some of the functions of the DAR and DBR.

The display file addressing system described above makes possible the storage of multiple display files of variable length and

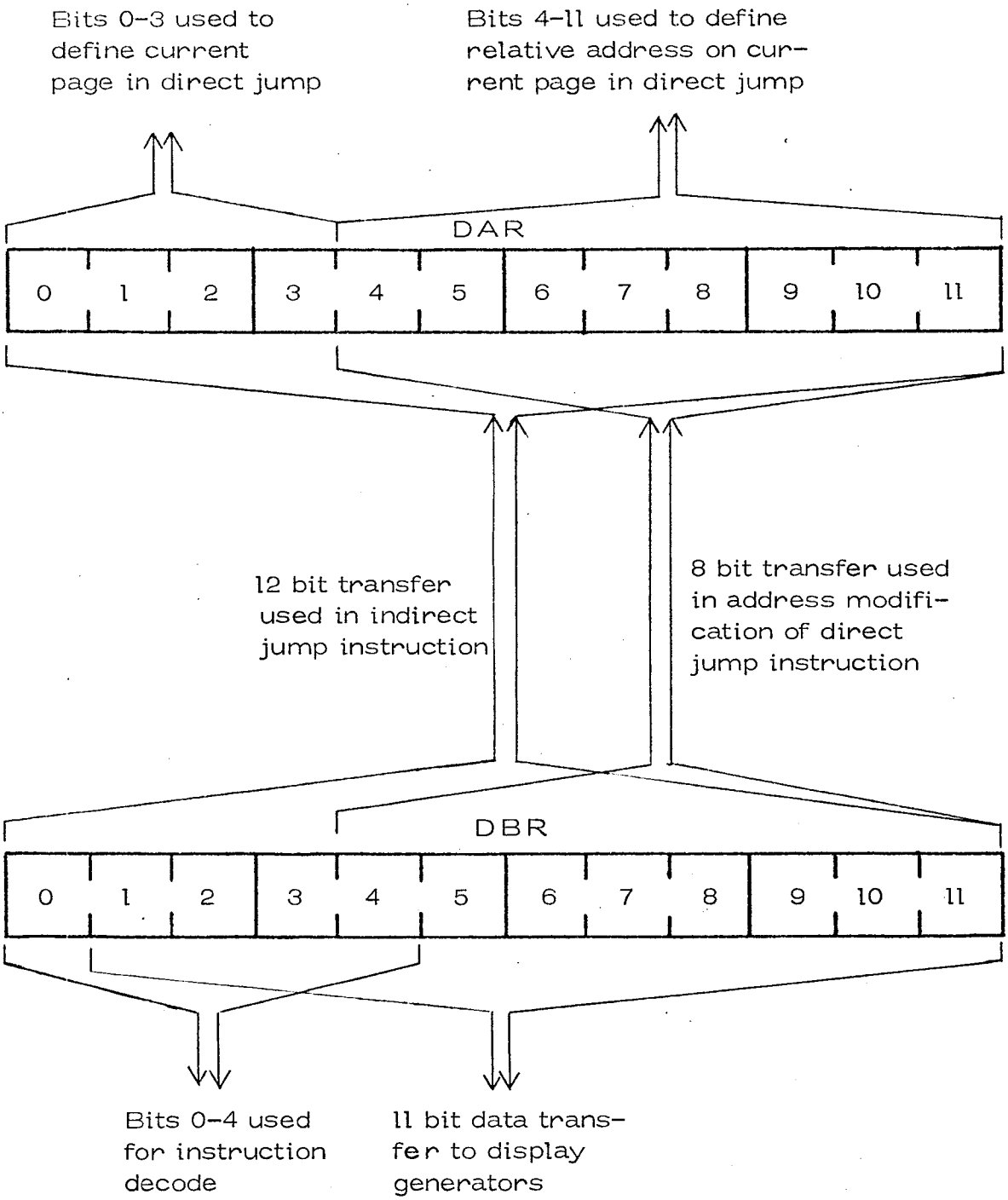


FIGURE I

DAR AND DBR BIT TRANSFER ARRANGEMENT

the selection of any of these display files under PDP8/S control. An example will best demonstrate this ability. Shown in Figure II is a typical memory map. Let us assume that four separate display files are stored in the refresh memory. To select a display file the PDP8/S need only deposit a pointer address in location 0001. That is, the address 002 deposited in location 0001 will select file number 1, address 0331 deposited in location 0001 will select file number 2, etc. This depositing of the pointer address need only be done once and the refresh memory will remain in the proper loop until another selection is made. Memory locations 3431 to 4096 are still available for additional display files or as space is available for additions or insertions to any of the display files as described earlier in this section.

To sum up the concept of display file refresh addressing, it is important to remember that: a data word is recognized by decoder logic which initiates controller action to increment the DAR by one memory location, a branch instruction is recognized by decoder logic which initiates controller action to modify the last eight bits of the DAR to access a location relative to the current page or page '0' in the case of a direct jump or changes the contents of the DAR completly to access an absolute location in the case of an indirect jump. Conceptual details of memory cycle timing and of PDP8/S access to the refresh memory are given in sections 2.2.5 and 2.2.6 respectively.

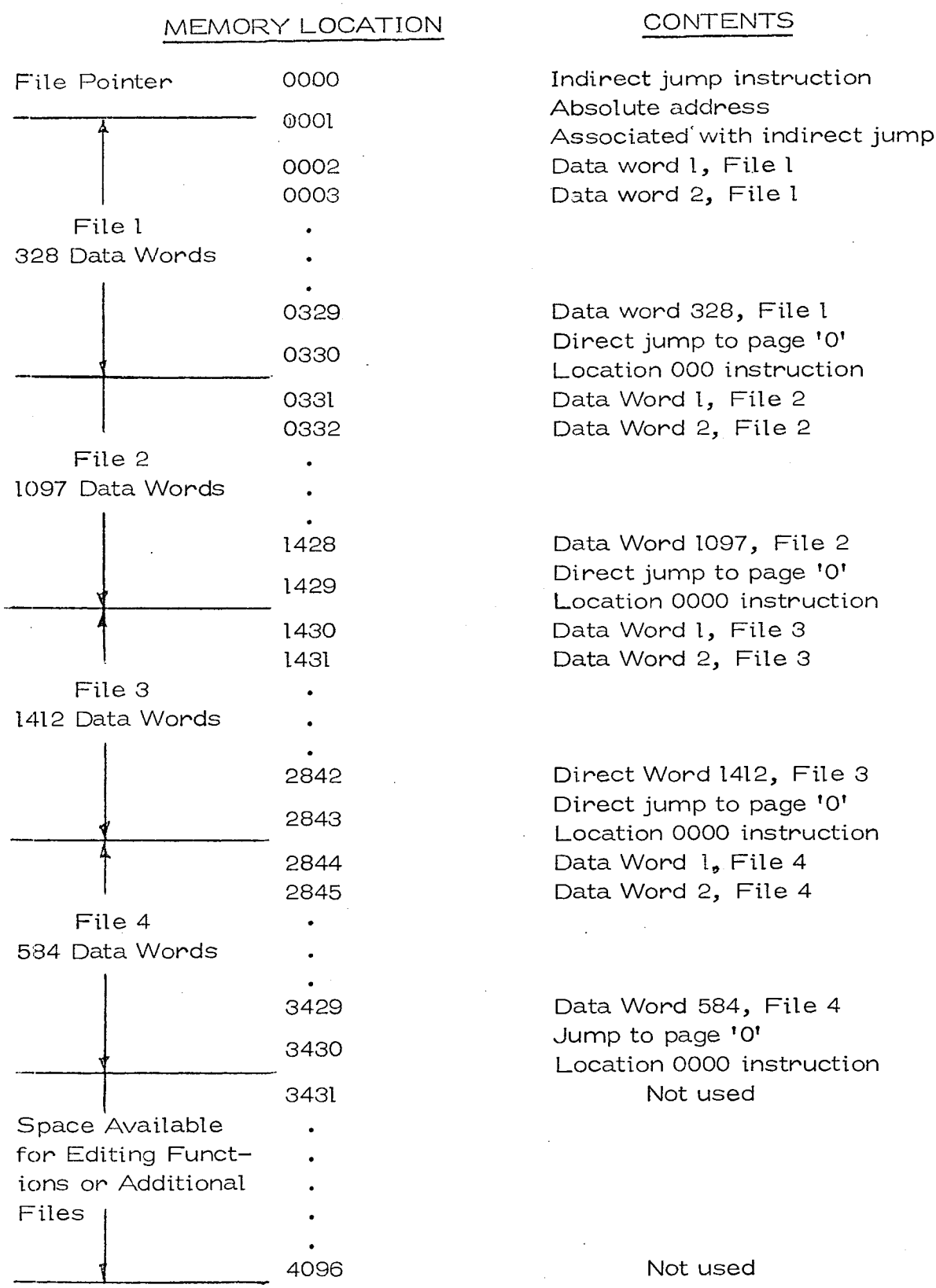


FIGURE II

DISPLAY MEMORY MAP

2.2.3 Data Channelling.

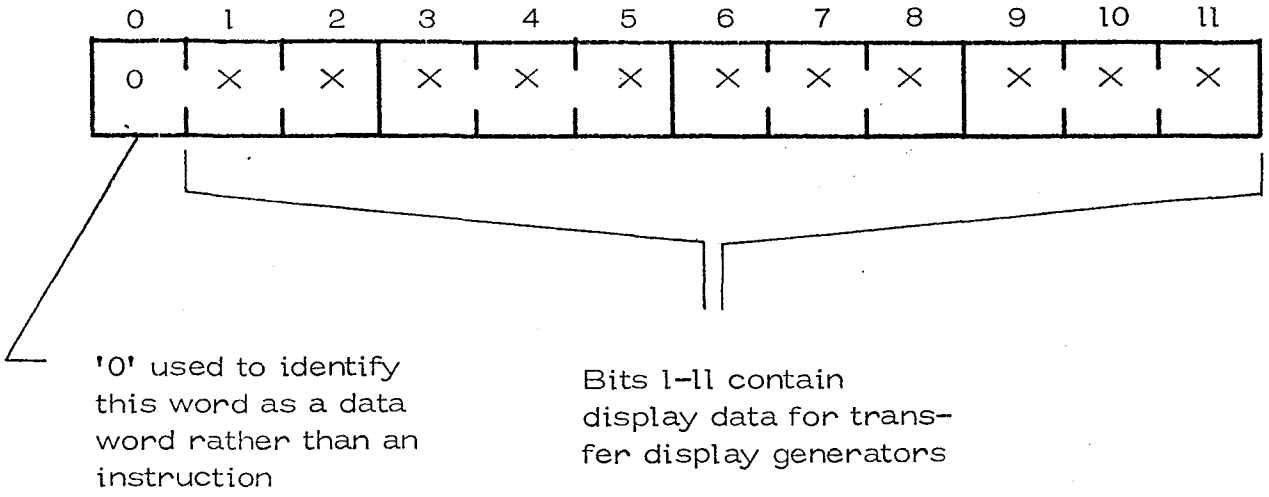
It was pointed out in Chapter I that a basic function of the data flow controller and refresh memory was to channel data to the appropriate display generators. This function could be implemented in two possible ways. The first approach considered was to set aside a number of bits in the data word itself that could be used to make a piece of data recognizable to decoder logic as destined for a particular display generator. For the three display generators already under development this would require the use of two bits of the data word for the necessary decoding. If more display generators were to be added at a later date, additional data word bits would have to be reserved. Since any word put out from the memory also had to be recognized as data or instruction (see section 2.2.2.) which requires the use of one bit, the total number of bits of a data word that could not be used to pass display information to a display generator would be a minimum of three or 25% of the available information bearing space. The second approach was to prefix any block of data (of any length) with an instruction embedded in the display file which would indicate to the controller logic that all data immediately following and up to the reading of a new data channelling instruction word was to be channelled to the display generator indicated by the prefixing instruction word. Since data destined for a particular display generator would, in general, be arranged in blocks or groups in the

in the display file, and since it was undesirable to further shorten the length of the data word, the second method of data channelling was adopted. Although this method of data channelling increases the number of non-data words in a display file it more than compensates with an increased information carrying capability of data words.

2.2.4 Display File Word Format.

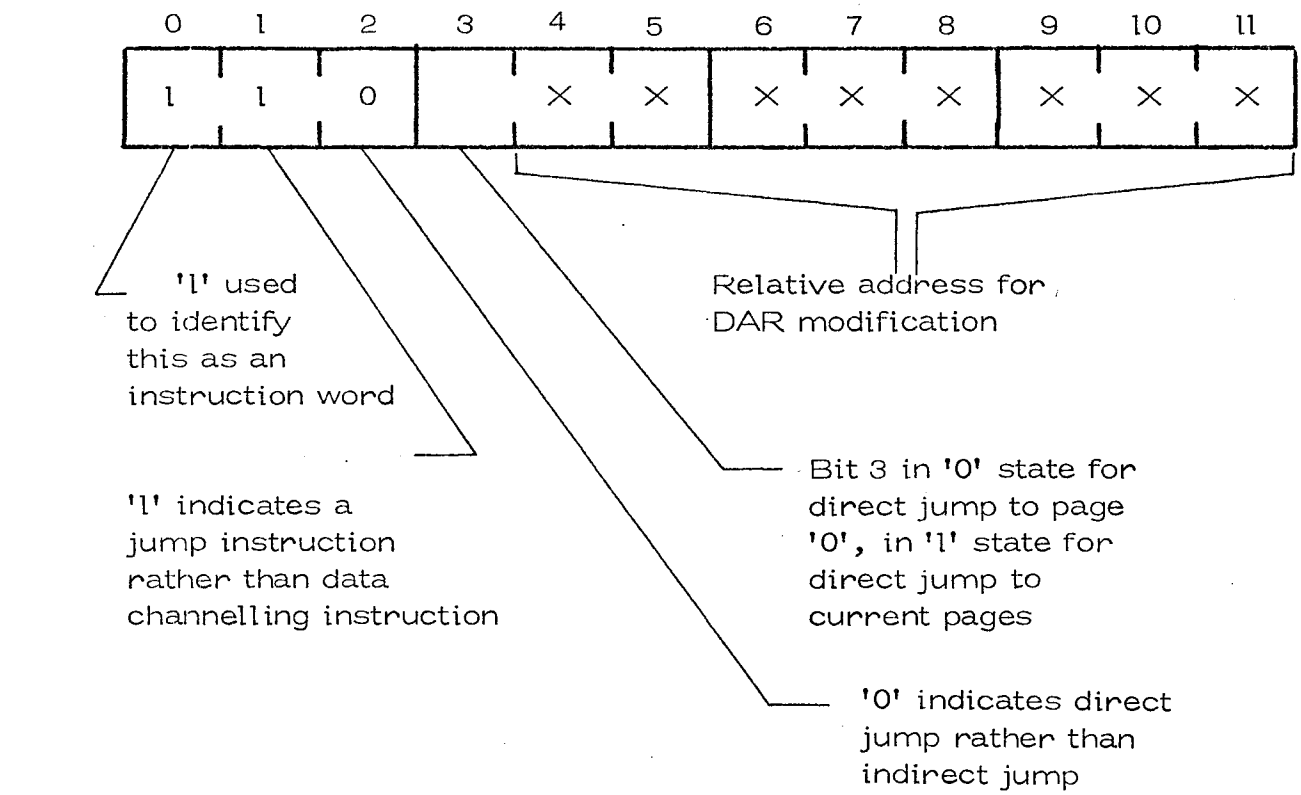
Sections 2.2.2 and 2.2.3 outlined the use of three types of words that may appear in a display file; the data word, the jump instruction, and the data channelling instruction. The following word formats and instruction codes were chosen for programming convenience.

Data word format is shown below:

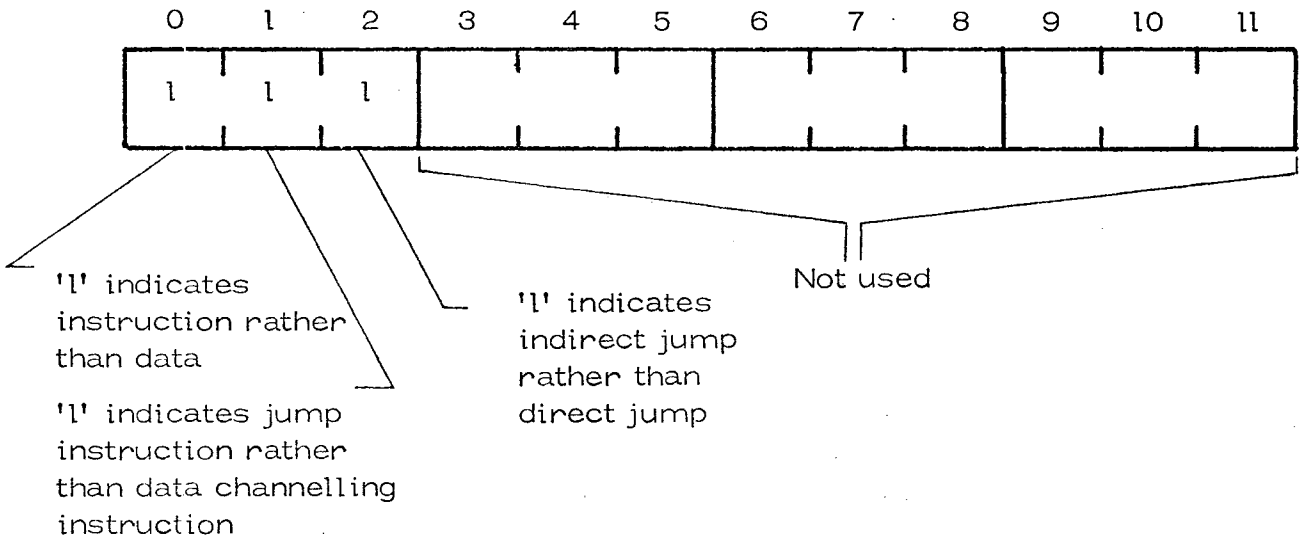


The following formats are those of jump instructions:

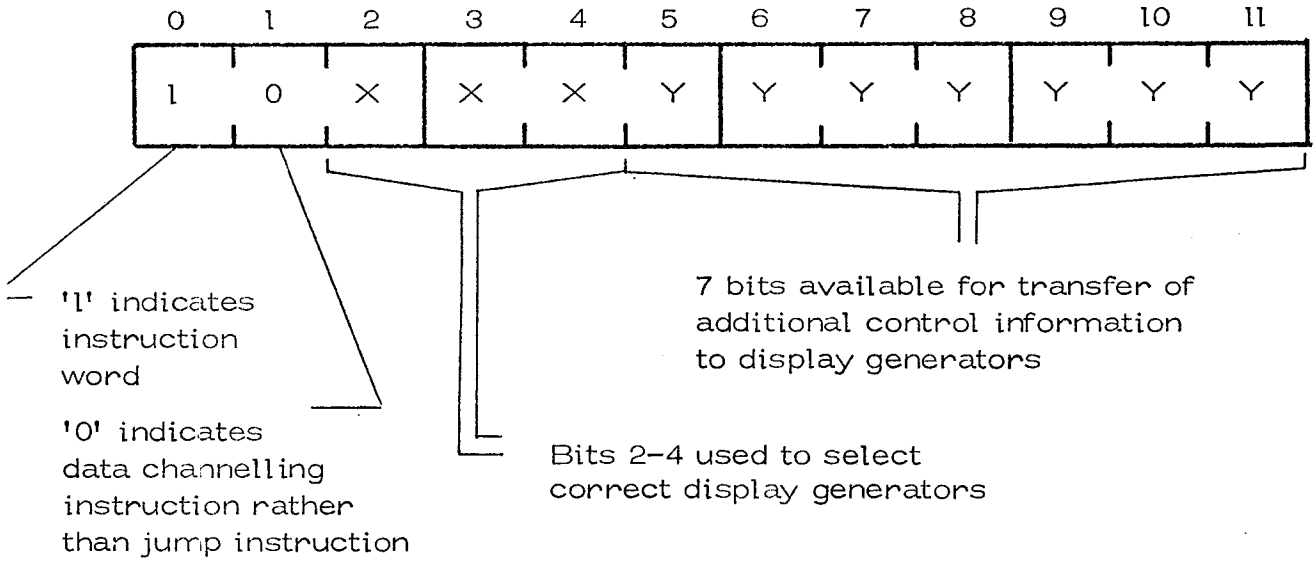
DIRECT JUMP



INDIRECT JUMP



The format below is that of data channelling instructions



2.2.5 Memory Cycle Initiation and Display Refresh Timing.

As was pointed out in Chapter I, one of the problems involved in the design of a CRT refresh system was that of feeding data to display generators that operate with a variety of timing requirements. It was also pointed out in section 2.1 that the use of a random access core memory could overcome this difficulty by allowing each display generator to operate at its own speed. This section outlines the concepts of the system used to implement display timing.

The outstanding advantage of using a random access core memory in any situation where timing may become complex is its ability to perform a memory cycle only on command from some external source. In this case the device which initiates a memory cycle is the data flow controller (DFC). The data flow controller is an arrangement of logic and flags that decides which memory location to access, when to access the memory, and what to do with the memory word when it arrives. The logic and detailed operation of the data flow controller are covered in Chapter III. In this section the data flow controller will be considered as a logic block with certain inputs and outputs. The interaction between the data flow controller and the PDP8/S are described in section 2.2.6 but for now it need only be kept in mind that the PDP8/S initiates the very first display

refresh memory cycle and from that initial start the DFC governs all refresh memory cycles.

To achieve display refresh timing one signal line from each display generator was required by the DFC. The state of this signal line indicates to the DFC whether that display generator is currently in generation or sitting idle and ready to accept data. In return, the DFC supplies a signal line to each display generator which is used to instruct the generator to read data from the DBR and begin generation.

Once again, the best approach to describing refresh timing is through the use of an example. Refer to Figure III which is a block diagram of the refresh timing arrangement and to Figure IV which is a small but very typical display file. First, assume that the PDP8/S has initiated the very first memory cycle which is a read from location 0000_8 into the DBR, the value of the DAR and DBR being initially 0000_8. As the contents of memory location 0000_8 arrives at the DBR, the first five bit positions are decoded and this word is recognized as an indirect jump instruction. The DFC is signalled the presence of an indirect jump instruction. The DFC immediately signals the DAR count logic to increment the DAR by one and then it initiates another memory read cycle. Due to the actual internal timing of the H201 memory stack (see section 3.2) the decoding, incrementing of the DAR, and the initiating of a new memory cycle

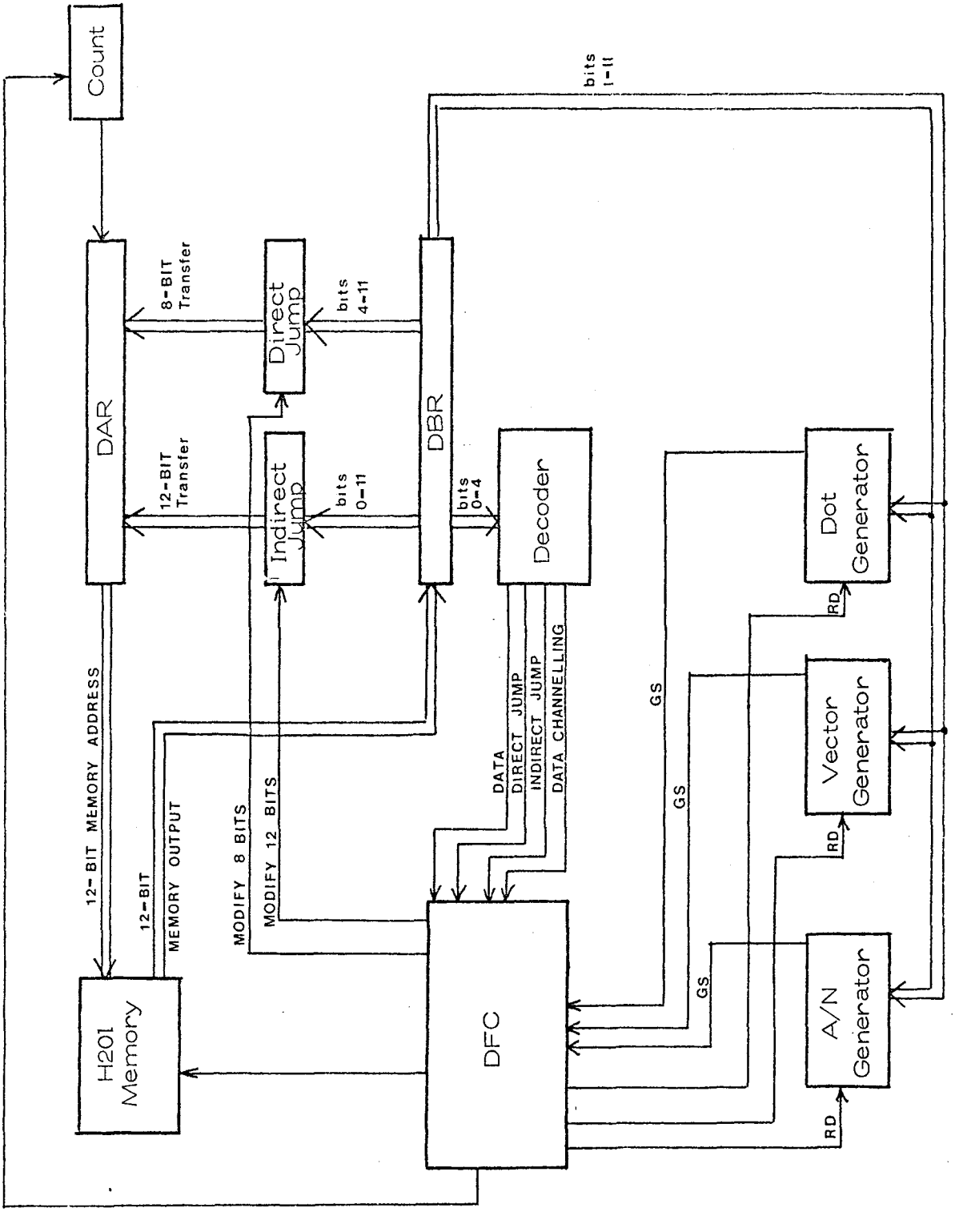


FIGURE III
DISPLAY REFRESH TIMING
BLOCK DIAGRAM

GS - Generator Status
RD - Read Data

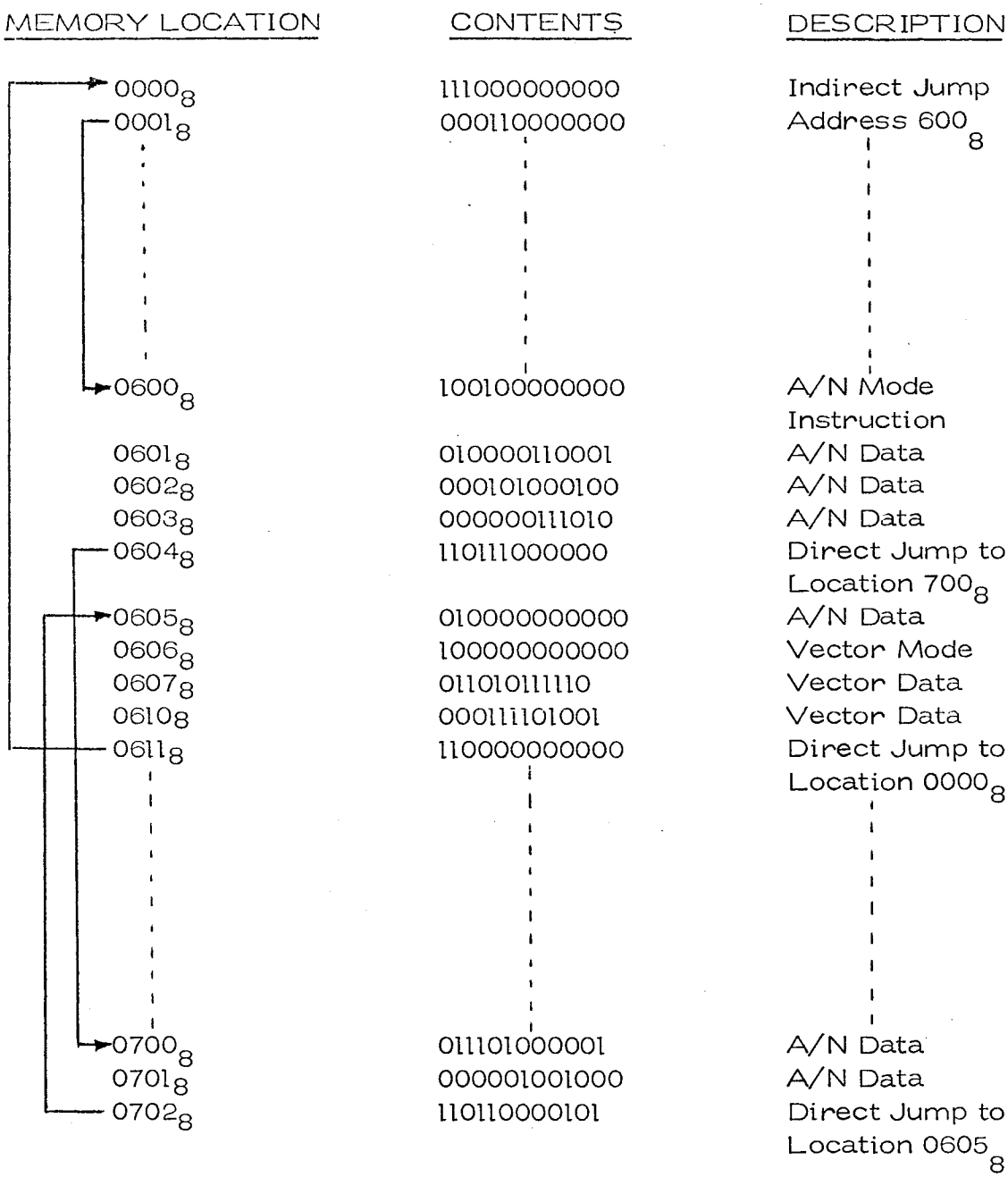


FIGURE IV

TYPICAL DISPLAY FILE

is all complete before the completion of the first 8 μsec memory cycle. Thus the second memory cycle begins immediately after the first with no unnecessary time loss. As the contents of memory location 0001_8 arrives in the DBR, the DFC "remembers" that this word is a new address to be used for modification of the DAR and so inhibits any decoding of this word which may have any bit configuration and thus cause an erroneous decoding. The DFC then signals the indirect jump logic to transfer the contents of the DBR to the DAR so that the DAR now contains the address 0600_8. The DFC then initiates a new memory cycle which again follows the preceding one with no loss of time. As the contents of location 0600_8 arrives in the DBR, the word is decoded as an A/N data channelling instruction or an A/N "mode" instruction. The DFC then sets a flag to ensure that all subsequent data words will be channelled to the A/N display generator. The DAR is then incremented and another memory cycle initiated again with no unnecessary time loss. The contents of location 0601_8 arrive in the DBR and are decoded as a data word. The DFC immediately senses the A/N generator status line which at this time indicates that the A/N generator is idle and ready to accept data and a command is immediately issued to the A/N generator to read the data word. The DAR is incremented and another memory cycle is immediately initiated, again no time is lost between memory cycles. The contents of location 0602_8 arrive in the DBR and are decoded as data. The DFC

senses the A/N generator status and finds that the generator is still busy with the last piece of data. The DFC then increments the DAR but does not initiate a new memory cycle. Instead, the DFC continuously senses the A/N generator status and holds the data in the DBR. As soon as the A/N generator status indicates to the DFC that the A/N generator is finished, a command is issued to the A/N generator to read the data already waiting in the DBR. A new memory cycle is then initiated. Location 603_8 is read and also contains data and the last cycle is repeated. Location 0604_8 contains a direct jump instruction which is decoded upon arrival in the DBR and the DFC is signalled. The DFC immediately signals the direct jump logic to modify the last eight bits of the DAR so that the DAR now holds the address 0700_8 and a new memory cycle is initiated which follows immediately upon the last one. Location 0700_8 is thus immediately read and decoded as data. The DFC senses the A/N generator status line and finds that the A/N generator is still busy with the data that was read from location 0603_8. The DFC then increments the DAR and waits for the A/N generator to finish. (From preliminary research on the display generators[3] it was estimated that the A/N generator would have the lowest cycle time of the three generators and that this cycle time would have a minimum in the neighbourhood of 20 μsec. Thus, the refresh memory could perform two eight μsec memory cycles before the A/N generator finished one character. This allows a branch to

be executed and a new piece of data to be available in the DBR by the time the A/N generator is ready for more data). Immediately following the indication from the A/N generator status line that the A/N generator is ready to accept more data the DFC issues a command to the A/N generator to read the data from location 0700_8 already waiting in the DBR. The remainder of the display file shown in Figure IV is processed by the DFC in the same manner as described above.

If the last piece of information in the display file (location 610_8) had been destined for the A/N generator, there would be a slight delay before the first piece of data in the display would be directed to the appropriate generator on the second pass (next refresh) due to the fact that four instruction memory cycles (locations 0611_8, 0000_8, 0001_8, 0600_8) are necessary between the first data cycle and the last. If we consider the A/N generation time as 20 μsec then the delay would be approximately 12 μsec (that is 32 μsec the time required for four memory cycles minus 20 μsec A/N generation time). However, in the example of Figure IV the last data word is vector mode data and as the vector generator would likely take considerably longer to process a word of data, there would not likely be any delay in the data flow to display generators.

The most notable features of the refresh timing scheme described above are:

(i) There is very little, if any, time wasted in which a display

generator must sit idle while waiting for the refresh memory to deliver data. This means that any display generator with a minimum cycle time greater than eight usec will be supplied data as fast as it can process that data for CRT display with very few exceptions.

(ii) Each display generator sets its own pace so there are no restrictions placed on the design of the timing of any display generator. Even if the display generator cycle time is shorter than that of the memory it will be serviced at a lower than maximum rate but with no timing difficulties.

2.2.6 PDP8/S Interaction with DFC and Refresh Memory Logic.

Up to this point very little has been said about the communication between the PDP8/S and the CRT display equipment. This section deals with the mechanics of the PDP8/S control over the CRT display facilities.

The PDP8/S controls the CRT display system in two ways. The first type of control is its ability to start up and shut down the operation of the system. Starting up requires an initialization of all display flags and display registers (the flip-flops of which may come on in random states at power up) and initiation of the first display refresh memory cycle (that is a read from location 0000_8 into the DBR) as described in the previous section. Shutting down is achieved by inhibiting all memory refresh cycles and resetting all display

registers and flags. Thus, shutting down is, in fact, realized by reinitialization. The second type of PDP8/S control of the CRT display is by controlling the contents of the refresh memory both by writing in the original display files prior to the initiation of CRT display operations and by modification additions, insertions, and deletions of material in the refresh memory during the time the CRT display system is actually in operation. This includes the type of data file selection described in section 2.2.1. As was outlined in the design criteria, it would be desirable if the second mode of control could be achieved without disturbing the running of the CRT display refresh (that is, without temporarily shutting down the refresh memory to make necessary changes).

Before going into more detail about the PDP8/S access to the display refresh memory it is worthwhile to go into a brief description of PDP8/S I/O facilities although for a more thorough and comprehensive coverage the reader should refer to the Digital Equipment Corporation publication 'Small Computer Handbook'.[4] The PDP8/S is equipped with a bus type I/O facility which includes input to and outputs from its accumulator register, peripheral device selection lines, a program interrupt line, and an I/O skip line. Peripheral device selection is achieved by external decoding of six device selection lines, the state of which are set by the I/O instruction being executed by the PDP8/S. Once a device is selected it receives any combination of

three sequential IOP pulses also determined by the I/O instruction (discussed later in this section). All data communications between the PDP8/S and the CRT display equipment is in the form of 12-bit words transferred from and received by its accumulator (AC). A signal on the Program Interrupt (PI) will cause the PDP8/S to branch to some predefined subroutine. The I/O skip line is used to inform the PDP8/S a peripheral device is ready to accept data or instructions or whether a peripheral has performed an assigned task. The PDP8/S remains in an I/O loop included in its program sequence for the purpose of waiting on peripherals until the I/O skip line (IOS) signals it to branch out of the waiting loop and continue processing.

If the PDP8/S was to have access to the refresh memory it was necessary that it be provided with access to two registers similar to the DAR and DBR. One register must hold the address of the location that is to be accessed and the other to hold the data to be written into the memory or receive the data read from the memory. It was conceivable that the DAR and DBR themselves could serve this purpose and so serve a double role. However, this would require that refresh be halted and the information held in these registers be destroyed each time the PDP8/S accessed the memory. This would necessitate a reinitialization and restart for each PDP8/S interrupt of refresh. Instead, it was decided to provide two registers designated Memory Address Register (MAR) and I/O Buffer (IOB) for this purpose. These

two registers would be connected to the I/O bus of the PDP8/S and each would be equipped to accept data from the PDP8/S AC and the IOB would be equipped to transfer data to the PDP8/S to facilitate PDP8/S reading of the memory as well as writing into the memory. Thus, the PDP8/S under program control could output a 12-bit address to the MAR and a 12-bit display file word to the IOB and then issue a command to write the word held in the IOB into the memory location specified in the MAR. Similarly the contents of a refresh memory location specified in the MAR could be read into the IOB and then transferred into the PDP8/S AC. During periods when the CRT display would not be in operation, these transfers would be no problem as the refresh memory could service the PDP8/S exclusively. However, for refresh memory updates during refresh operation, timing of PDP8/S memory access cycles presented a problem. Obviously, since the refresh memory could service only one at a time, some type of timing system had to be developed that would allow the CRT display circuits and the PDP8/S I/O to share access to the refresh memory. As dictated by the design criteria it was desirable to disturb or delay the CRT refresh functions as little as possible during such memory access sharing so as to ensure that a maximum possible amount of display data could be refreshed with a flicker free refresh rate. This job of refresh memory access coordination was to be another function of the DFC. Once again, the

advantages inherent in the use of a random access core memory made possible a solution to the problem of dual access timing with minimum interference to the refresh function. Since the H201 memory stack has a relatively short cycle time of 8 μsec and since the shortest display generator cycle time was to be of the order of 20 μsec (actually this time turned out to be 18.5 μsec), it was apparent that during refresh of strings of data there would be periods of time in which the refresh memory would sit idle while the DFC waited for display generation to finish (refer to the example in the previous section). These idle periods would be more than long enough for the H201 memory to perform a memory cycle in service of the PDP8/S. Thus, the DFC could initiate the "stealing" of a cycle from the regular refresh duties of the refresh memory to accomplish a PDP8/S I/O memory cycle. These stolen cycles would not effect the regular refresh timing of the display generators nor would they destroy the bit configurations currently residing in the DAR and DBR. In addition, since the minimum time required for the PDP8/S to perform an I/O program loop or sequence is 102 μsec (as pointed out in Chapter I), it would be, in almost all cases, very probable that the DFC would be able to "find" an opportunity to steal a cycle before the PDP8/S could be ready to request another memory access cycle. Thus, the PDP8/S would not be materially delayed in processing I/O to the refresh memory due to the regular refresh duties of the memory.

To implement such a system of dual memory access, the PDP8/S would have to make its request for a memory access cycle to the DFC which would hold the request until it found an opportunity to initiate the cycle and then notify the PDP8/S through the I/O ship line that the requested cycle had been completed. It should be noted here that the PDP8/S would remain in an I/O skip loop[4] incorporated in the program under execution by the PDP8/S until the IOS caused the PDP8/S to branch out of the loop, but since the DFC would likely perform the I/O task before the PDP8/S could make another request, this I/O skip loop may be left out of PDP8/S programs at the discretion of the programmer.

The possibility that at some future date a display generator would be developed with a cycle time less than eight μsec presented the problem that the DFC would not be able to find sufficient opportunities to cycle steal on behalf of the PDP8/S and so substantially slow down PDP8/S I/O functions to the CRT display system. It was therefore decided to implement a "computer override" option that would be selected under program control of the PDP8/S and would allow the PDP8/S to take priority over the display refresh for display memory access.

Since a memory may be accessed by only one set of registers at a time, the above dual access system necessitates that the MAR-IOB and DAR-DBR register combinations be equipped with logic that would

allow them to be switched into or out of access to the memory address decode, the memory inhibit, and memory output lines of the H201 memory stack. This selection of registers for memory access would be under DFC control.

The dual access system implies a set of priorities associated with memory access. Before detailed design of the DFC could start, this priority system had to be laid down more explicitly. The highest priority had to be a memory cycle already in execution as it would be disasterous to initiate a new memory cycle while the memory was already in mid-cycle. The following list represents the memory access priority scheme in descending order of priority.

(1) A memory cycle already in execution.

(2) A PDP8/S memory cycle requested while computer override option is in force.

(3) A display refresh memory cycle.

(4) A PDP8/S memory cycle requested while computer override option is not in force.

Due to the fact that data transfers between the refresh memory and the PDP8/S would often take the form of blocks written into sequential locations in the refresh memory, it was felt that an automatic increment facility incorporated into the MAR would provide more rapid data transfer and reduce program steps. This "auto index" facility would be implemented by making the MAR a counter

register and by having the DFC initiate a one's count in the MAR every time a read or write cycle was requested by the PDP8/S and successfully completed. This would be an option selected under program control. Thus, a starting address could be placed into the MAR from the PDP8/S I/O bus and the auto index option then enabled by a single I/O command (the option would have to be disabled during the transfer of a 12-bit address from the PDP8/S due to the logic of a counter register, see Appendix II) and all subsequent memory cycles would access memory locations in sequence from the starting address until the option is disabled by I/O command.

Before a list of the I/O commands developed for the PDP8/S interaction with the CRT display system is given, a brief discussion of the PDP8/S I/O command format and operation will be useful.

A typical PDP8/S I/O command word has the format shown in Figure V. Bits 0-2 make up the operation code and are decoded in the PDP8/S to initiate an I/O instruction sequence. Bits 3-8 make up the device selection code. Each peripheral device must have in its circuitry at least one W103 DEC standard device selector module[4] (DS) which is comprised of a 12-input and gate (in actuality this gate is a NAND gate but for the sake of continuity it may be considered as an AND gate) the output of which enables, through DCD gates (see AppendixII) three pulse amplifiers (PA) as shown in Figure VI. Bits 3-8 and their compliments are available on the I/O bus and they are

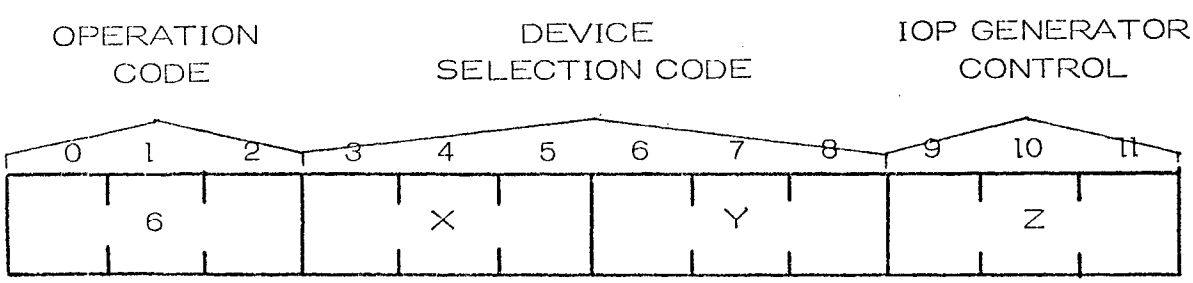


FIGURE V
TYPICAL I/O INSTRUCTION

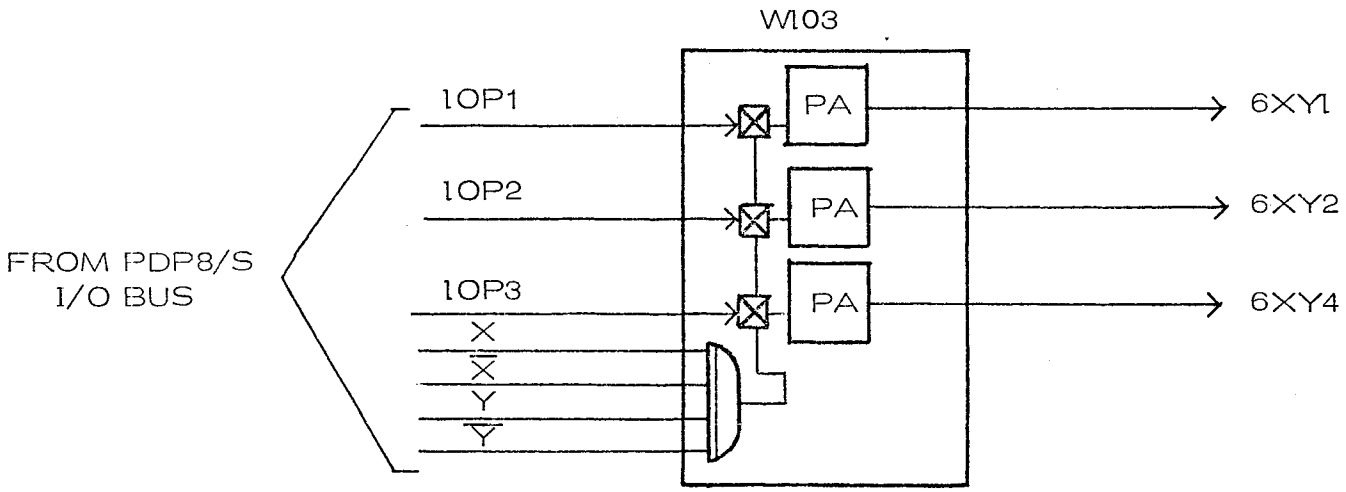


FIGURE VI
W103 MODULE CONFIGURATION

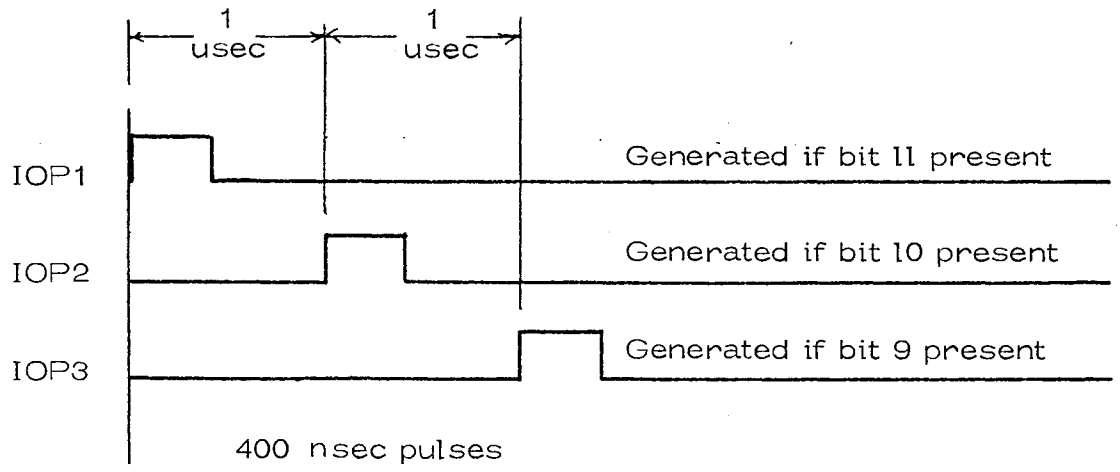


FIGURE VII
PDP8/S IOP PULSE TIMING

applied to the inputs of all DS AND gates. Each DS AND gate is arranged so that it will "turn on" only with a specified 6-bit code from bits 3-8. Thus for each I/O command only one DS is enabled. Bits 9-11 control the generation of the IOP pulses. These pulses are generated in the I/O logic of the PDP8/S and are available on the I/O bus if they are selected in the I/O instruction word. Figure VII gives the timing of the three pulses and indicates that an IOP pulse will be generated only if the corresponding bit is specified in the instruction word. Thus, the I/O instruction of the PDP8/S serves the purpose of applying any combination of three possible pulses to a specific device among its peripherals. These pulses may be used for triggering data transfers or any variety of control functions.

The following list gives the commands developed for the PDP8/S interaction with the CRT display equipment and the associated codes. It should be noted that the first section of the list gives the basic commands (that is; the commands in which only one IOP pulse is called for so that each of these commands initiates only one function), the second part of the list gives commands that are combinations of two or three of the commands in the first section of the list (that is; commands where more than one IOP pulse is specified so that two or three of the functions of the basic commands are performed by one instruction word). DS codes were selected to facilitate the combining of several functions into one command where this was feasible.

	I/O Instruction Code	Description and Comments
Part I		
1	6721.....	Clear IOB
2	6724.....	Write from IOB into Mem. Log. Specified in MAR.
3	6722.....	Load IOB from PDP8/S AC.
4	6734.....	Read from Refresh Mem. Location Specified in MAR into IOB
5	6751.....	Skip if Requested Operation not Complete.
6	6754.....	Transfer Contents of IOB into PDP8/S AC.
7	6731.....	Clear MAR
8	6732.....	Load MAR from PDP8/S AC.
9	6741.....	MAR Auto Index Enable On.
10	6742.....	MAR Auto Index Enable Off.
11	6711.....	Initialize Display or Shut Down.
12	6712.....	Computer Override Option On.
13	6714.....	Computer Override Option Off.
14	6701.....	Initiate First Display Refresh Memory Cycle.
Part II		
15	6723.....	Clear and Load IOB from PDP8/S AC. Combination of 1 and 2.
16	6725.....	Clear Memory Location Specified in MAR. Combination of 1 and 3.
17	6727.....	Load Memory Location Specified in MAR from PDP8/S. Combination of 1, 2, and 3.

	I/O Instruction Code	Description and Comments
18	6733	Clear and Load MAR from PDP8/S AC. Combination of 7, 8.
19	6737	Read From Memory Location Specified in PDP8/S into IOB.

A total of six DS modules were used in the actual implementation of the DFC and refresh memory logic. This provides the possibility of using eighteen basic commands and as only fourteen were used, four commands were left for the possible implementation of future logic such as the light pen system described in the next section.

2.2.7 Possiblities for Future Implementation of a Light Pen.

It was pointed out in Chapter I that one of the design criteria for the DFC and CRT refresh memory was the potential for addition, at some future date, of a light pen for a higher level of man-machine interaction. As a light pen implementation was not part of the original design problem, no detailed logic designs were attempted. However, the following conceptualization of such a system will serve to demonstrate that the system of data flow control and CRT refresh conceptualized in the preceding sections of this chapter does have the potential for light pen implementation and so successfully fulfills that particular design criterion.

A light pen consists of some type of light sensitive electronic

device (a light sensitive transistor for instance) mounted in a convenient fixture for holding, generally, in the same manner as a ball point pen. The end of the light pen holding the light sensitive device is applied directly to the face of the CRT screen. When the CRT beam travels across the screen and excites phosphors directly under the light sensitive head of the light pen, a pulse is produced by the light sensitive device and its associated circuitry. This pulse is used to initiate some type of computer or controller action. In this case, the light pen would be placed on a CRT screen area where one of the display generators was producing a CRT trace. When the beam causes this area to produce a "burst" of light during the cycling refresh of the display, it would be desirable to use the resulting pulse to initiate action to secure and hold the refresh memory address of the data word that was being processed to produce that particular trace. This data word address could then be transferred to the PDP8/S by a program sequence entered when the PDP8/S received a PI, also initiated by the light pen pulse. The PDP8/S could then process this data word address for editing, software option selection (menus) or any number of other software applications.

The functions described above could be implemented by the introduction of a new 12-bit register, possibly designated Data Address Tracking Register (DATR). This register would hold the address of any piece of data currently in generation. Therefore, at

any time when a command is issued from the DFC to a display generator to read data and begin generation, a simultaneous command would be issued to the DATR to copy from the DAR the 12-bit address currently held by the DAR. The DAR itself could not double as the DATR since, as was previously described, the DFC may execute several instruction cycles while a data word is being processed by a display generator, the DAR may no longer hold the address of the data currently in generation when a light pen pulse is received by the DFC. In actuality, since the DAR is incremented before a command is issued by the DFC to the display generators, the address in the DATR would always be one location ahead of the address of the data in generation. This, however, could easily be taken into account by software associated with the light pen operation. With the DATR operating as described above, when a light pen pulse arrives at the DFC logic, the DFC would immediately inhibit the transfer of address from the DAR to the DATR, thus "freezing" the desired address in the DATR. Simultaneously, the DFC would initiate a PI to the PDP8/S causing it to jump to a predefined subroutine that would transfer the contents of the DATR, the outputs of which are connected to the I/O bus inputs to the PDP8/S AC, into the AC. The PDP8/S could then process this address according to its software. After processing, the PDP8/S would have to issue an I/O command to the DFC to discontinue its inhibit of DAR to DATR transfers. Thus, the system would be

ready for more light pen interrupts.

The light pen system described above is purely hypothetical and was not implemented. Therefore, it will be left out of the block diagram and summary included in the next section.

2.2.8 System Block Diagram and Summary

Figure VIII gives a block diagram of the overall system. Appendix I gives the key to Figure VIII. In the next chapter, where applicable, data and command lines will be labelled with the codes shown in Figure VIII and described in Appendix I to achieve a clear correlation between the system block diagram and actual system logic.

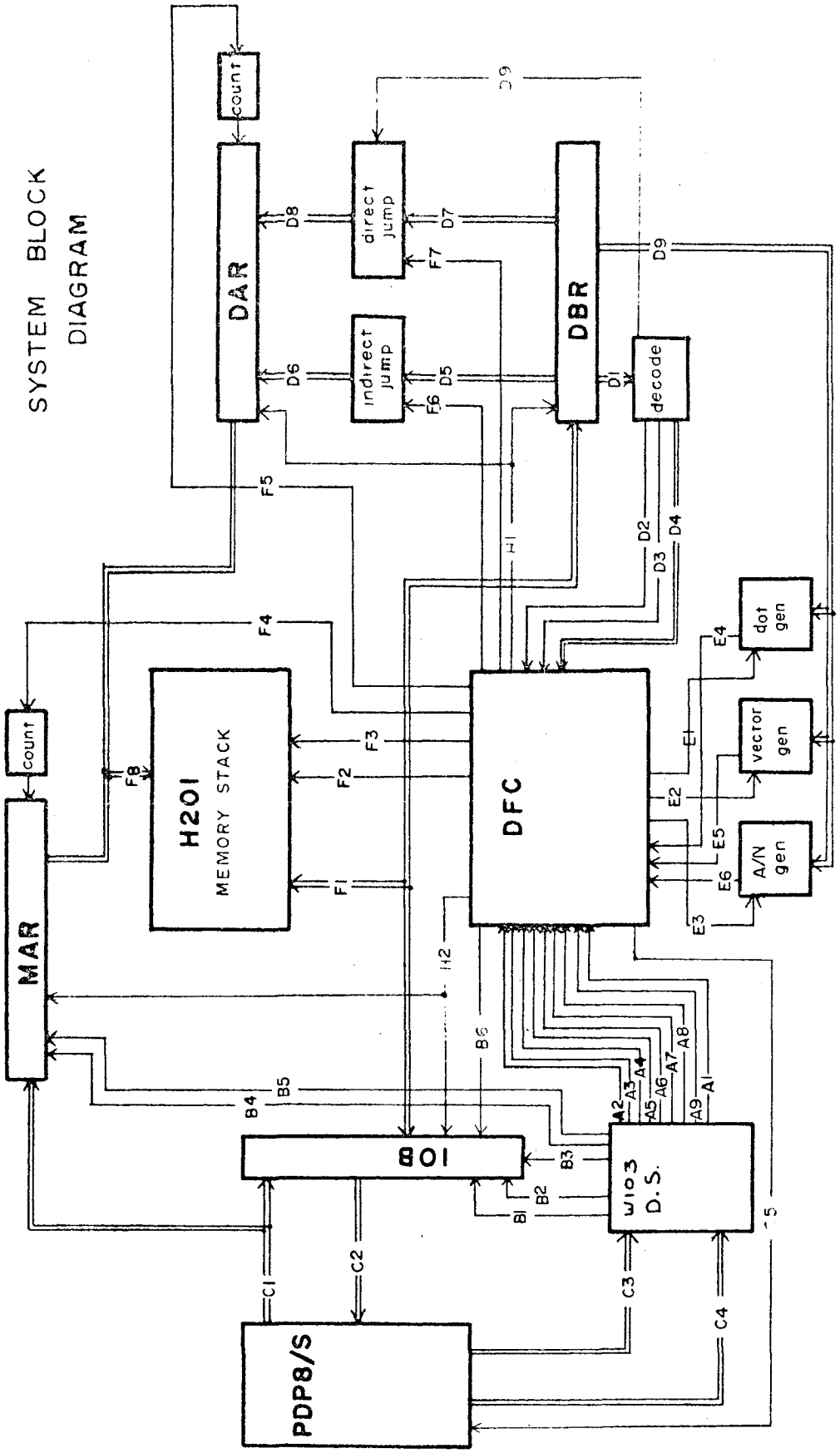
The system described in this chapter was conceived to satisfy the design criteria specified in Chapter I. However, the most notable overall feature of this system is its ability to maintain data flow control and high speed refresh leaving the PDP8/S processing capability free for more complex software functions. This system could provide a theoretical maximum of:

$$(1/30 \text{ sec})\ (18.5 \times 10^{-6} \text{ Sec}) \simeq 1800$$

words of data to an A/N generator running with a word processing time of 18.5 μsec as was the projected processing time of the A/N generator under development. The calculation of this number ignores time in which the display refresh is delayed by instruction execution by the DFC but this delay will be very small or may not exist at all

FIGURE VIII

SYSTEM BLOCK DIAGRAM



due to the fact that the DFC can execute commands while "waiting" for the display generator to finish execution. Thus, this system can produce a CRT display equivalent to a full printed page of this thesis (again only considering A/N data) compared with the absolute maximum of 5 lines that the PDP8/S could sustain in a flicker-free display. In addition, the processing ability of the PDP8/S which would be very seriously hampered by having to keep up a display refresh, is free to process programs and subroutines designed to make full and active use of CRT display, for which only updating of display files need be done. This means that the PDP8/S can maintain a large CRT display and also oversee other I/O devices and software that use the CRT display.

CHAPTER III

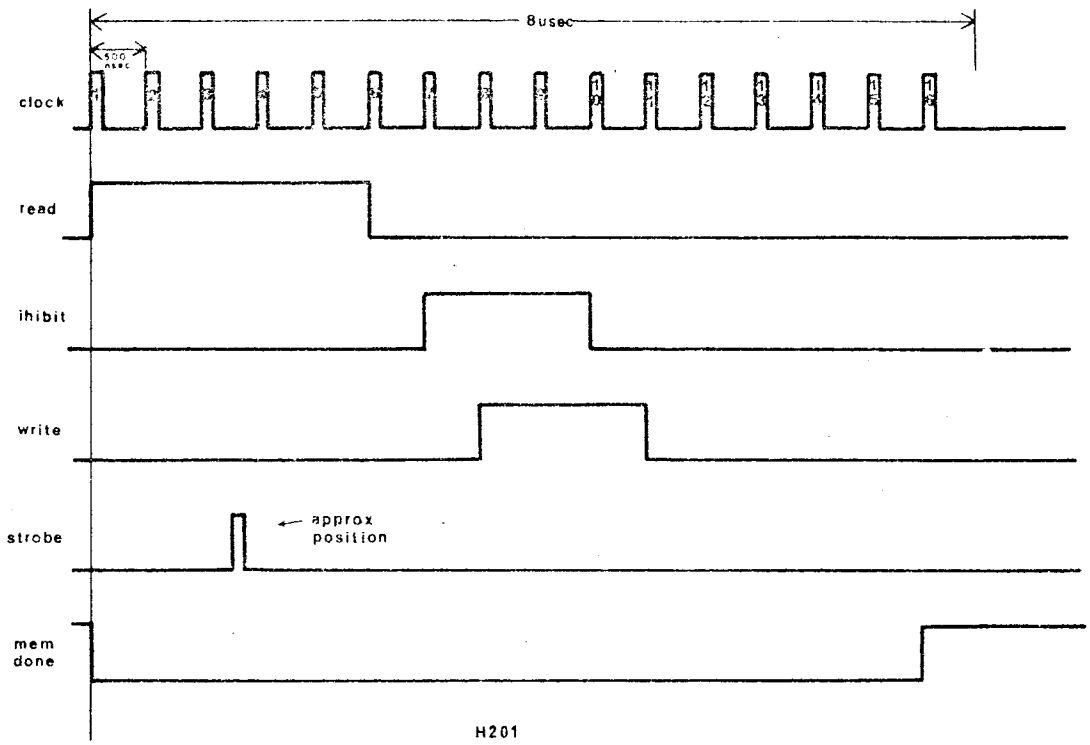
SYSTEM LOGIC

3.1 Introduction

This system was designed for implementation with a combination of Diode Transistor Logic (DTL) and Diode Capacitor Logic (DCDL). An examination of Appendix II describing logical functions and symbolism should be made before study of this chapter.

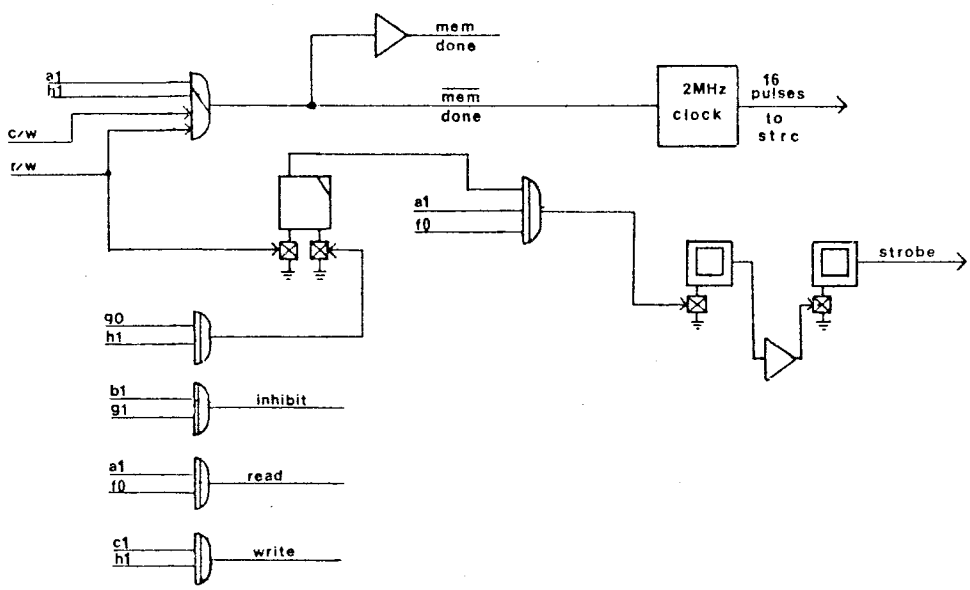
3.2 Memory Timing and Control

No attempt will be made to describe random access core memory theory as this subject is well covered both in the general case[5] and in specific for the H201 memory stack[6]. However, a discussion of the timing and control curcuits for the H201 memory is essential as this circuitry is integrated into the DFC timing logic. The timing of the 8 μsec H201 memory cycle is shown in Figure IX. This cycle is based on 16-100 nsec pulses spaced at 500 nsec intervals. This train of 16 pulses is received from a 2MHz clock which is enabled by the timing control logic to pass 16 pulses upon request for a memory cycle and then disabled. As shown in Figure IX the READ level, which enables the read drivers of the memory stack, is 2½ μsec in duration turning on at pulse 1 and off at pulse 6. Similarly, the WRITE and INHIBIT level timing is shown. The strobe pulse (STROBE) is



H201
MEMORY CYCLE
TIMING

FIGURE IX



MEMORY TIMING
LOGIC

FIGURE X

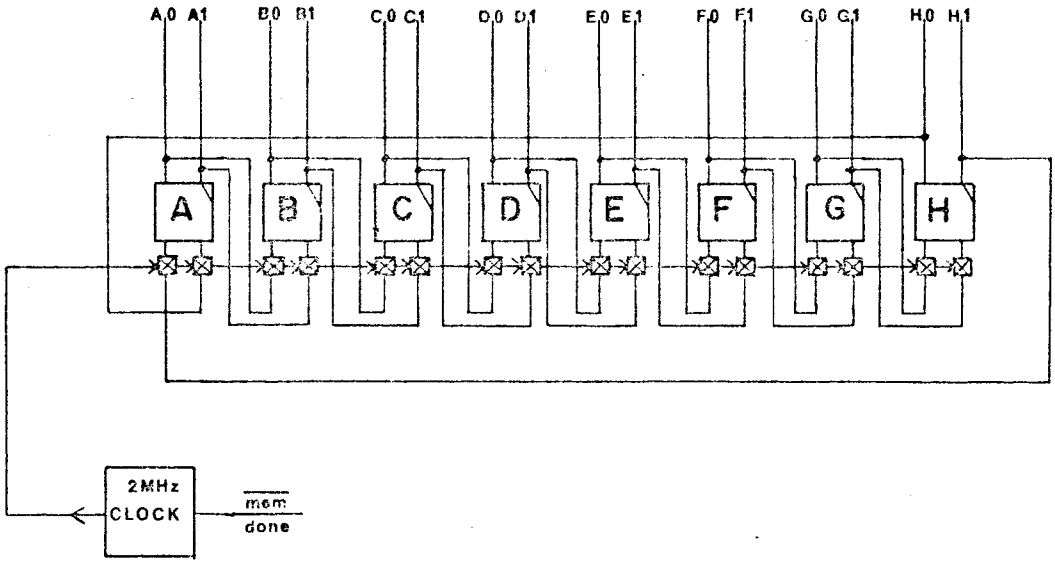


FIGURE XI

SWAP TAIL RING COUNTER

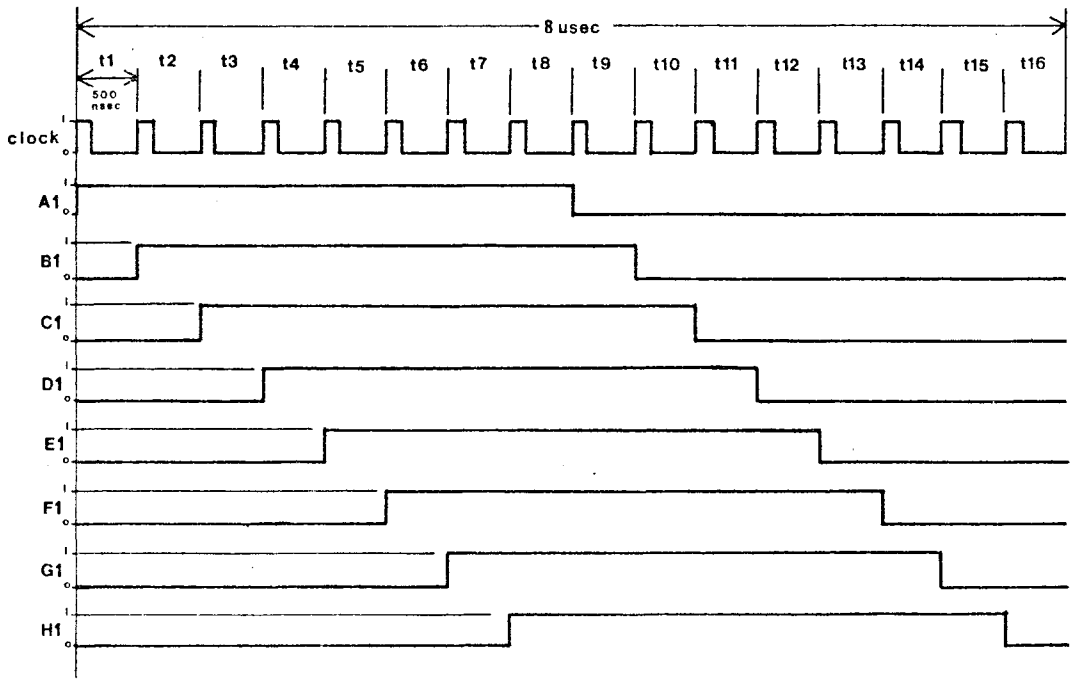


FIGURE XII

STRC TIMING

positioned approximately as shown but its position and duration are subject to memory tuning.[6] The 3 μsec unused portion of the cycle is allowed for memory driver and sense amplifier recovery (it should be noted here that the address bits applied to the memory address decode inputs must remain stable throughout the read and write time but may be changed or switched out at pulse 11). The importance of understanding the H201 memory cycle lies in the generation of exactly 16 clock pulses and the READ, WRITE and INHIBIT levels. These timing functions are controlled by the states of a Swap Tail Ring Counter (STRC)[6] which receives the 16 clock pulses as shift pulses. The STRC is a ring counter, in this case an 8-bit ring counter, with the connections between the last bit and the first bit reversed. Figure XI shows the STRC used in this application. Figure XII gives the timing diagram of the "1's" output of each stage for an input of 16 shift pulses from the clock. The '0' output timing diagram is just the compliment of that shown in Figure XII. The resulting configuration of the STRC for each of the 16 steps is shown below along with the positioning of the READ, WRITE and INHIBIT levels.

PULSE		LOGICAL STATE OF STRC STAGES							
		A	B	C	D	E	F	G	H
1		1	0	0	0	0	0	0	0
2		1	1	0	0	0	0	0	0
3	READ	1	1	1	0	0	0	0	0
4		1	1	1	1	0	0	0	0
5		1	1	1	1	1	0	0	0

PULSE		LOGICAL STATE OF STRC STAGES								
6			1	1	1	1	1	1	0,	0	
7		(	1	1	1	1	1	1	1	0	
8	INHIBIT	(	1	1	1	1	1	1	1	1	) WRITE
9		(	0	1	1	1	1	1	1	1	) WRITE
10			0	0	1	1	1	1	1	1	)
11			0	0	0	1	1	1	1	1	
12			0	0	0	0	1	1	1	1	
13			0	0	0	0	0	1	1	1	
14			0	0	0	0	0	0	1	1	
15			0	0	0	0	0	0	0	1	
16			0	0	0	0	0	0	0	0	END OF CYCLE

The READ, WRITE and INHIBIT levels are generated by gating the outputs of the STRC stages. For example, the READ level is generated by the gating of A1 "AND" F0. That is:

$$READ = A1.F0 \quad \ldots\ldots\ldots\ldots \quad (1)$$

Similarly

$$WRITE = C1.H1 \quad \ldots\ldots\ldots\ldots \quad (2)$$

$$INHIBIT = B1.G1 \quad \ldots\ldots\ldots\ldots \quad (3)$$

This can be verified by inspection of the above table and Figure XII. The timing control logic is shown in Figure X (it should be noted that in all subsequent figures all lines labelled with an alphabetic character either upper or lower case from A-H followed by a "1" or "0" and unbracketed refer to the outputs of the STRC stages and should not be confused with the lines shown in Figure VIII which will be shown for reference but bracketed for identification). The line in Figure X

labelled MEM DONE is used to signal that the memory is not currently in cycle. The level MEM DONE is used to enable and disable the clock. For a proper cycle to take place the clock should be enabled upon arrival of either a clear-write cycle request (C/W) or a read-write cycle request (R/W) (clear-write cycle refers to the clearing of a memory location and then writing into it, a read-write cycle refers to the reading of a word from a memory location into an external register and then writing the word back in so the contents of the location is preserved) and should stay enabled until 16 shift pulses have been sent to the STRC. This is done by the gating of C/W "OR" R/W "OR" A1 "OR" H1. That is:

$$\overline{\text{MEM DONE}} = \text{R/W} + \text{C/W} + \text{A1} + \text{H1} \quad \ldots\ldots\ldots (4)$$

Implimentation of the above equation allows the clock to be enabled upon the arrival of a C/W or R/W pulse and to remain enabled as long as either stage A or stage H of the STRC is in the "1's" state which allows exactly 16 pulses to be generated before the clock is disabled. The MEM DONE level goes to the "1's" state at the end of the cycle and remains that way until a new cycle is initiated by C/W or R/W. A R/W request requires the generation of the STROBE pulse. As shown in Figure X, the R/W pulse sets a flip-flop into the "1's" state and the "1's" output is gated with A1 and F0 to the DCD gate input of a "one shot" the output of which is buffered by an inverter to the DCD gate input of a second "one shot". These oneshots have an adjustable period

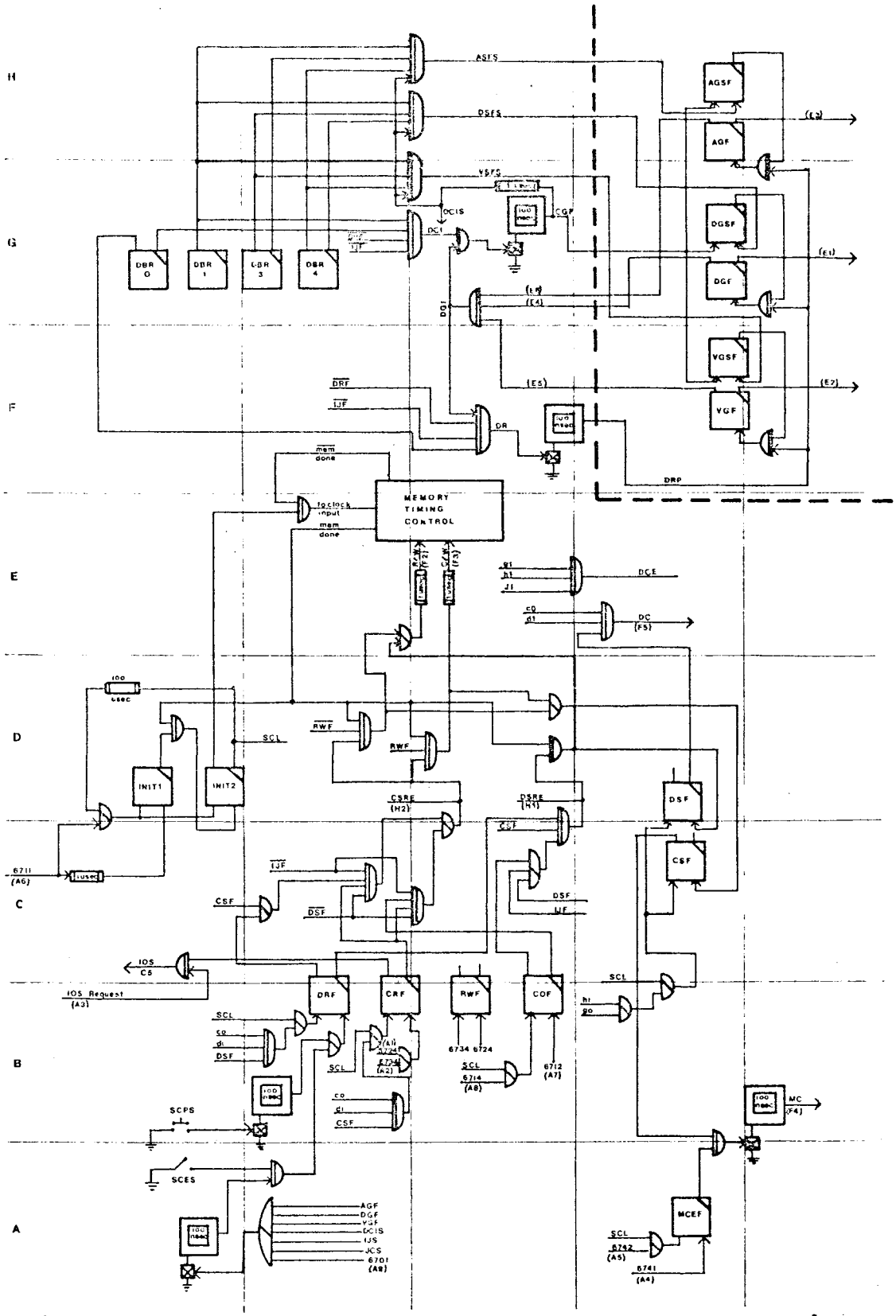


FIGURE XIII

DFC AND DECODER LOGIC-1

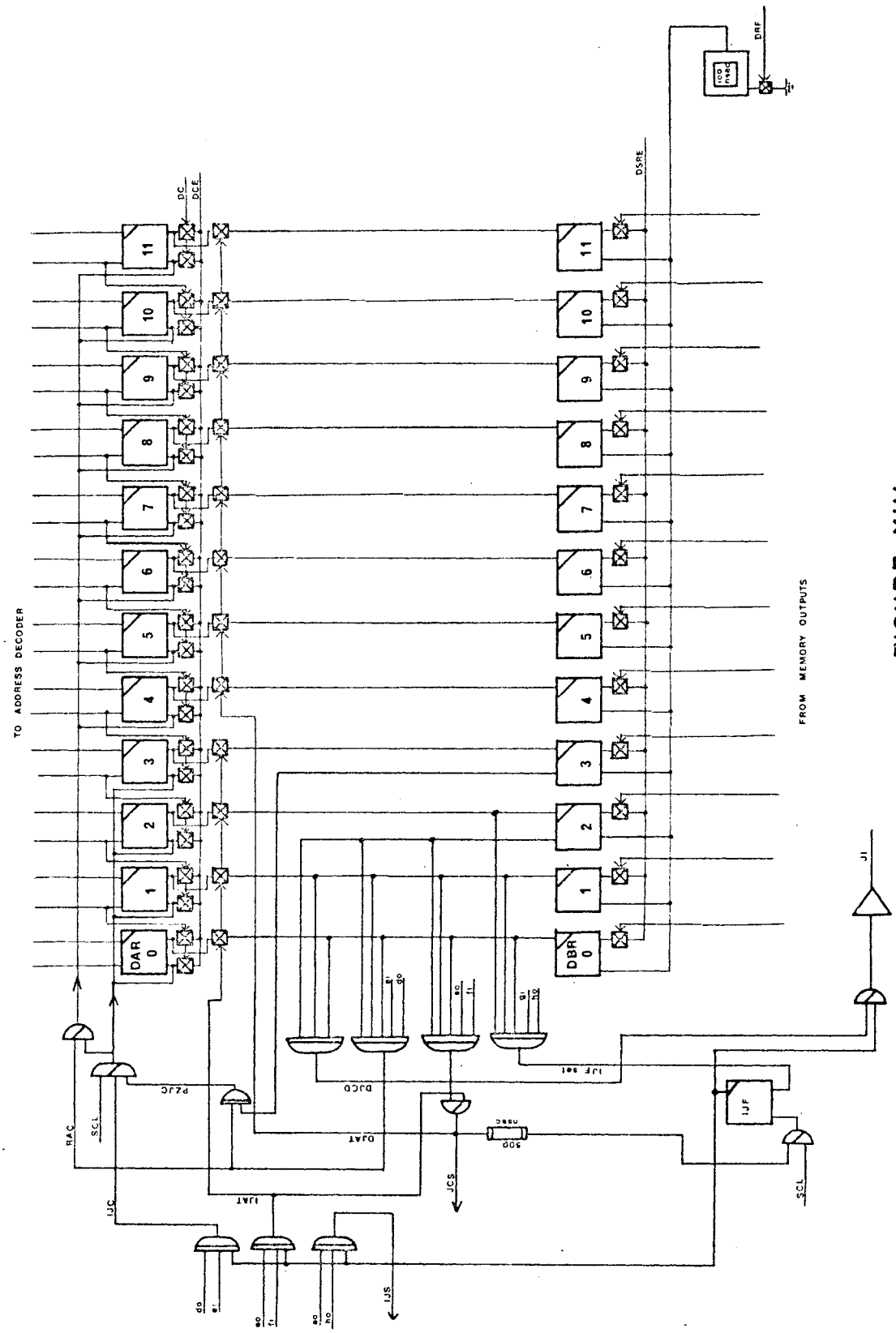


FIGURE XIV

DFC AND DECODER LOGIC-2

and are used to set the position and duration of the STROBE.

The above description of the memory timing and control logic serves to demonstrate the use of gating the STRC outputs to obtain accurate and dependable timing functions. The timing of many DFC functions is based on further use of the STRC output gating.

3.3 DFC and Decoder Logic

3.3.1 Notation and Figures

Since the DFC and DECODER logic are highly integrated, they will be described together and are shown in Figures XIII* and XIV. The configuration of the W103 Device Selector modules is not shown since this logic is a standard part of the PDP8/S I/O procedure and is well covered in the "Small Computer Handbook"[4]. The outputs of the DS modules are shown as command inputs to the DFC and these lines are labelled with the instruction codes listed in Chapter II, along with the corresponding designation assigned in Figure VIII. The H201 memory stack and its timing control logic will be shown as a single logic block.

3.3.2 Initiation/Shut Down Logic

As was mentioned in Chapter II, the functions of initiation, of the display system and of system shut down are implemented with the same logic. Both these functions are achieved by clearing

---

* Note that Figure VIII shows a grid reference system that will be used along with references to Figure VIII for easy reading.

all system flags and the DAR and DBR registers into the "0" state. Referring to Figure XIII-1C, the computer command requesting initiation or shut down is 6711 (A6). It is necessary that the system not be shut down during a memory cycle since memory contents may be destroyed. Two flags are used and designated INIT1 and INIT2 (Figure XIII-1D). As the 6711 arrives both INIT1 and INIT2 are cleared (this is necessary as they will come on in random states during power up). The 6711 pulse is then delayed by 1 μsec and used to set INIT1 into the "1's" state. The "1's" output of INIT1 is "ANDED" with MEM DONE and the output of this gate applied to the set terminal of INIT2. Therefore, INIT2 will go into the "1's" state only when the memory is not in cycle. The "1's" output of INIT2 is used to clear the system logic at initialization or shut down and this level is designated System Clear Level (SCL). The SCL is passed through a 100 nsec delay element and the delayed SCL is used to clear INIT1 and INIT2. Thus, the SCL is a 100 nsec level which is a long enough period to suppress signals coming back from the display generators requesting new memory cycles. The SCL level is shown in all the appropriate locations in Figure XIII and XIV and will not be discussed further.

3.3.3 Memory Cycle Initiation and Priority Logic

A memory cycle is initiated by the arrival of either a R/W or C/W pulse at the memory timing control logic. The logic used in the generation of R/W and C/W and priorities involved in memory

cycle requests is shown in Figure XIII. Seven flags are involved in governing memory cycle requests and they are:

Display Ready Flag (DRF) Figure XIII-2B ..... Used to signify that the CRT display system is ready for another memory cycle. This flag is initially set by PDP8/S command 6701 (A9) and is continually reset by DFC logic from signals received from display generators and instruction logic as discussed in Chapter II.

Computer Ready Flag (CRF) Figure XIII-2B ..... Used to signify that the PDP8/S has requested a memory cycle (either read or write). This flag is set by PDP8/S commands 6724(A1) and 6734(A2)

Read-Write Flag (RWF) Figure XIII-3B ..... Used to indicate whether a computer requested memory cycle is a read or a write cycle. This flag is set by PDP8/S command 6724 (A1) and cleared by 6734 (A2).

Computer Override Flag Figure XIII-3B ..... Used to indicate the selection of the computer override option. It is set by PDP8/S command 6712 (A7) and

cleared by 6714 (A8).

Display Service Flag (DSF) Figure XIII-4D ..... Used to indicate that a memory cycle is currently in progress in service of the CRT display refresh.

Computer Service Flag (CSF) Figure XIII-4C ..... Used to indicate that a memory cycle is currently in progress in service of a PDP8/S request.

Indirect Jump Flag (IJF) ..... Shown in Figure XIV is enabled when an indirect jump instruction is decoded in the DBR. This flag indicates that the next memory cycle must be a display service cycle to provide a new 12-bit address to the DAR.

To discuss memory cycle initiation it is best to start at the Computer Service Register Enable level (CSRE) and the Display Service Register Enable level (DSRE). These levels are marked in Figure XIII-3D and are used to switch MAR-IOB and DAR-DBR combinations into memory access (this will be discussed more fully in section 3.3.4). However, it is also these levels that cause the generation of R/W and C/W (Figure XIII-3E). Consider, firstly, that a display service memory cycle has been requested and that the DSRE has been switched from "0" to "1", DSRE is "ANDed" with MEM DONE and the output of this gate is used to set the DSF and is also applied to a 1 μsec

delay element through an "OR" gate (see Figure XIII-2A "OR" gate used since R/W may be requested by CRSE "OR" DSRE). The output of the delay is R/W (since the CRT display only requires read cycles no connection to C/W is necessary. As soon as R/W arrives at the memory timing control logic the clock is enabled and MEM DONE goes to the "0" state (see Figure X). Thus, R/W is a 1 μsec pulse. Since R/W is generated by "ANDING" of DSRE and MEM DONE the last of these two levels to arrive at the gate triggers the initialization of the cycle. Thus the memory must be idle before R/W can initiate a new cycle. The purpose of the 1 μsec delay element is a practical one. The DSRE and CSRE are used to switch register access to the memory, this requires the switching of 24 DTL gates and 12 DCD gates (see section 3.3.4) which considerably increases propogation delay in the electronics used to implement the system[8]. Therefore, a 1 μsec delay was used between the enabling of CSRE or DSRE and the generation of R/W or C/W to ensure that this memory access gating has settled before the memory cycle begins. Thus, the time for execution of a memory cycle is increased from 8 μsec to 9 μsec. A computer service memory cycle initiation is very similar except that CSRE is gated with RWF and CWF as well as MEM DONE to differentiate between a read cycle request and a write cycle request as shown in Figure XIII-2D+3D.

The priority scheme described in Chapter II is achieved

by the logic leading up to the enabling of CSRE or DSRE. During the system logic design, it was found advantageous to introduce a new priority level to accommodate the fetching of a new 12-bit address for the DAR during the memory cycle immediately following the encounter of an indirect instruction. This new priority made a display service memory cycle compulsory immediately after any display service memory cycle which happened to be an indirect jump instruction. This new priority is second only to a cycle already in progress for access to the memory. The following equations represent the generation of CSRE and DSRE from Figure XIII-2C+3C.

$$DSRE = (\overline{COF} + DSF + IJF) \cdot \overline{CSF} \cdot DRF \quad \ldots\ldots\ldots (5)$$

$$CSRE = ((CSF + \overline{DRF}) \cdot \overline{IJF} \cdot \overline{DSF} \cdot CRF) + (\overline{IJF} \cdot COF \cdot CRF \cdot \overline{DSF}) \quad \ldots\ldots\ldots (6)$$

Simplifying equation 6

$$CSRE = CRF \cdot \overline{DSF} \cdot \overline{IJF} \, (CSF + \overline{DRF} + COF) \ldots (7)$$

Before discussing these equations, note from Figure XIII-2B that DRF and CRF are cleared by the output "DSF . CO . 01" and "CSF . CO . D1" respectively. That is, the flags DRF and CRF indicating request for memory service are cleared only if their requested cycle has gone into execution and at the 11th clock pulse during that cycle, also note that CSF and DSF are cleared at H1 . G0 or the 15th pulse of any memory cycle (Figure XIII-4B). Examination of equations (5) and (7) shows:

(i) That DSRE can never become enabled while a computer service cycle is in progress (that is, while CSF is in the "1's" state). CSRE can never become enabled while a display service cycle is in progress (that is while DSF is in the "1's" state). Simply, once a memory cycle has begun it cannot be interferred with no matter how high the priority of the new request.

(ii) If COF and IJF are both in the "0" state, memory access becomes a matter of "first come first served" except that the display will be served in priority over the computer at times when simultaneous requests are residing in the DRF and CRF. This situation will occur most often when the DFC is processing instructions in the display file during which the DRF will be reset between pulse 11 and pulse 16 of the memory cycle. Thus, the CRF will have to "wait" until the DRF is left unset and then "steal" a cycle.

(iii) If COF is in the "1's" state there will be a suspension of all display service memory cycles, except when IJF is in the "1's" state in which case one display service memory cycle will be executed to satisfy and clear the IJF, and this suspension will remain in effect until COF is cleared by PDP8/S command.

3.3.4 H201 Memory Input and Output.

Figure XV gives the logic of the multi-accessing of

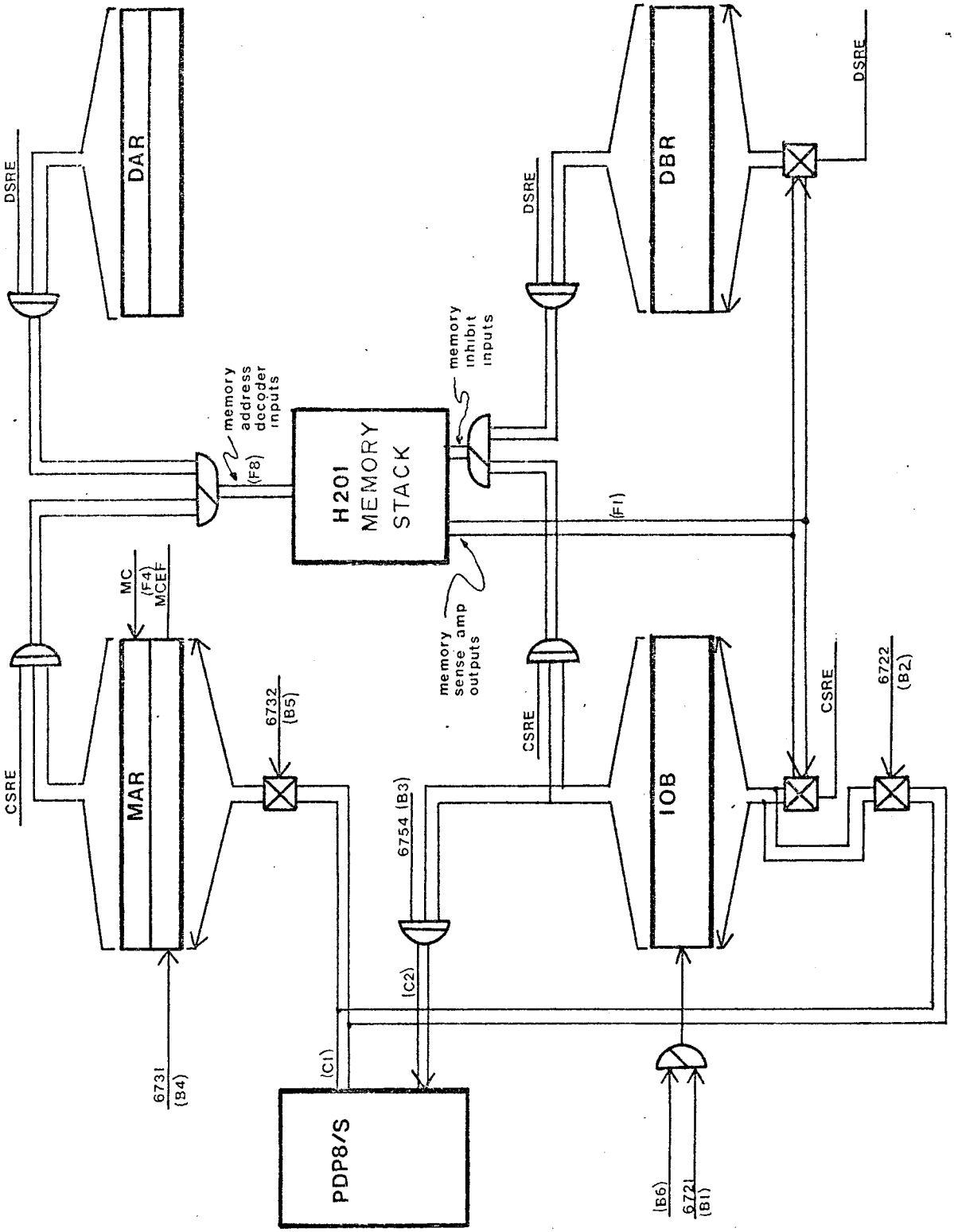


FIGURE XV

MEMORY I/O LOGIC

the memory. Input and output from PDP8/S accumulator is also shown along with the clear and count arrangement of the MAR-IOB combination. It should be noted that the memory address decode inputs are 24 bits (both "1's" and "0's" outputs of MAR "OR" DAR). This Figure shows how DSRE and CSRE switch the DAR-DBR and MAR-IOB combinations into and out of memory access. Since DSRE and CSRE can never be enabled simultaneously as shown in equations (5) and (7) only one register pair may access the memory at a time.

The clear and count logic of the DAR-DBR combination is shown in Figure XIV and described in the next section.

The count facility of the MAR is enabled by the MAR Count Enable Flag MCEF (Figure XII-4A) which is set by computer command 6741 (A4) and cleared by 6742 (A5). The MAR Count pulse (MC) is generated by the "ANDing" of MCEF with the '0' output of the CSF (Figure XIII-5B to trigger a 100 nsec one-shot to produce the MC shown also in Figure XV.

3.3.5 DFC Processing of Display Files.

As has previously been the case, description of the operation is best done through the use of an example. Shown below is a short display file that contains all the types of display file words that are encountered by the DFC and DECODER logic.

MEMORY LOCATION	CONTENTS	
$0[illegible]_8$	7000_8	Indirect jump instruction
$[illegible]3_8$	0200_8	DAR address word
.		
.		
.		
.		
0200_8	4400_8	A/N mode instruction
0201_8	XXXX	A/N data
0202_8	YYYY	A/N data
$0[illegible]3$	6000	Direct jump instruction

Consider that the system has been initialized and only awaits the arrival of a computer command 6701 to initiate the first refresh memory cycle. On arrival of the 6701 pulse setting the DRF through an "OR" gate as shown in Figure XIII-2A DSRE goes to "1" and initiates the first memory cycle as described in section 3.3.3. During the read portion of this cycle the contents of location 0000_8 arrives in the DBR. The first 4 bits are decoded by the logic shown in Figure XIV (review should be made of the display file codes discussed in Chapter II). The following sequence of events takes place during this cycle.

(1) At clock pulse 8, the DAR is enabled as a counter by the DAR COUNT ENABLE (DCE) level (Figures XIII-4E and XIV).

$$DCE = JI \cdot H1 \cdot G1 \quad \ldots(8)$$

where: JI is the Jump Indicator line.

and

$$JI = (DJCD \cdot IJF) \ldots\ldots(9)$$

where: DJCD is the DIRECT JUMP COUNT DISABLE line.

Therefore,

$$DCE = (DJCD - IJF) \cdot H1 \cdot G1 \quad \ldots\ldots \quad (10)$$

By equation (10) the DAR count facility remains enabled until clock pulse 15 (See section 3.2).

(ii) At clock pulse 11 the DAR is incremented by a pulse on the DAR COUNT (DC) line (Figures XIII-4E and XIV). Equation (11) governs the generation of this pulse.

$$DC = CO \cdot D1 \cdot DSF \quad \ldots\ldots\ldots\ldots \quad (11)$$

Thus, the incrementation of the DAR is done only during a display service memory cycle and after the portion of the memory cycle during which the memory address decode inputs (see Figure XV) must be held stable.

(iii) Also at clock pulse 11 the DRF is cleared as shown in Figure XII-2B.

(iv) At clock pulse 15 the IJF SET line is enabled and sets the IJF as is shown in Figure XIV.

(v) Also at clock pulse 15 the DSF is cleared as shown in Figure XII-4B.

(vi) At clock pulse 16 the DRF is reset by the Indirect Jump Set (IJS) line where:

$$IJS = AO \cdot HO \cdot IJF \quad \ldots\ldots\ldots\ldots\ldots \quad (12)$$

Therefore, the first memory refresh cycle is completed leaving the DRF set to initiate the second cycle (which must be a display service cycle since IJF is in the "1" state) upon the arrival of MEM DONE, described in section 3.3.3

The DRF and MEM DONE both go to the "1" state at clock pulse 6 and hence a new cycle begins immediately. Note that MEM DONE is cleared by the DRF through a 100 sec one-shot (see Figure XIII-3J) so it is ready to accept the contents of memory location 0001 during the read portion of the second cycle (actually at the time of the strobe). This word would normally be decoded as data since MBR00 is in the "0 state. However, since the IJF is set, the decoder does not treat this word as data. That is, the Data Read (DR) line does not become enabled (Figure XIII-3F). Equation (13) governs the DR line.

$$DR = DRF \cdot IJF \cdot (AGF \cdot DGF \cdot VGF) \quad \ldots\ldots\ldots (13)$$

where:

AGF is the A/N in Generation Flag (Figure XIII-4H)

DGF is the Dot in Generation Flag (Figure XIII-4G)

VGF is the Vector in Generation Flag (Figure XIII-4F)

The following sequence of events takes place during the second cycle.

(1) At clock pulse 8 the DAR fails to become enabled as a

counter since the Jump Indicator line (JI) is in the "0" state (see equations (8) and (9).

(ii) At clock pulse 11 the DRF is cleared.

(iii) At clock pulse 12 the DAR is cleared by the Indirect Jump Clear (IJC) line (Figure XIV) governed by equation (14).

$$IJC = DO \cdot E1 \cdot IJF \quad \ldots\ldots\ldots\ldots\ldots\ldots (14)$$

(iv) At clock pulse 13 the contents of the DBR is transferred to the DAR by a pulse on the Indirect Jump Address Transfer (IJAT) line (Figure XIV) governed by equation (15).

$$IJAT = EO \cdot F1 \cdot IJF \quad \ldots\ldots\ldots\ldots\ldots\ldots (15)$$

(v) Also at clock pulse 13 the pulse on the IJAT line initiates the setting of the DRF by enabling the Jump Complete set (JCS) line through an "OR" gate (Figure XIV). In addition, the JCS line is delayed by 500 nsec and the delay output clears the IJF (the 500 nsec delay ensures sufficient duration of the IJAT pulse).

(vi) At clock pulse 15 the DSF is cleared.

Thus, at the end of the second memory cycle the DAR holds the new address 0200_8, the IJF has been cleared and the DRF has been set ready to initiate a third memory cycle immediately upon the arrival of MEM DONE.

Again the third memory cycle follows immediately after the second (excluding the 1 μsec delay for DSRE gating functions) and the DBR was cleared at clock pulse 13 of the second cycle (after its contents had been transferred to the DAR), when DRF was set by JCS. During the read portion of the third memory cycle the contents of location 0200_8 arrives in the DBR and is decoded as a data channelling instruction and specifically as an A/N MODE instruction. Figure XIII-G1, G2, G3, H1, H2 H3 shows the decoding and implementation logic of data channelling instructions. It should be noted that the logic shown surrounded by a dashed line is part of the display generator logic and is physically located within the display generators. Since the AGF, DGF and UGF would be present in the display generators, it was decided to put them to double use and use them to implement some DFC logic. The following sequence takes place during the third memory cycle.

(i) At clock pulse 8 the DAR is enabled as a counter as described during the events of the first cycle.

(ii) At clock pulse 11 the DAR is incremented, as described for the first memory cycle, to hold the new address 0201_8.

(iii) Also at clock pulse 11 the DRF is cleared.

(iv) As the DRF is cleared its "0" output causes the Data Channelling Instruction line (DCI) to become enabled, governed by equation (16).

$$DCI = \overline{DRF} \cdot \overline{IJF} \cdot DBR00 \cdot \overline{DBR01} \quad \ldots\ldots\ldots (16)$$

The AGF, DGF and VGF are all in the "0" state, since none of the display generators has been active. Therefore, the Display Generator Idle (DGI) is enabled. The DCI is "ANDED" with the DGI to trigger a one-shot producing a 100 nsec pulse designated Clear Generator Flags (CGF). The CGF is used to clear three flags; the A/N Generator Select Flag (AGSF) (when this flag is set the A/N display generator will read data from the DBR upon the arrival of a Data Read (DR) pulse described during the next cycle), the Dot Generator Select Flag (DGSF) (same function for the Dot display generator, the Vector Generator Select Flag (VGSF) (same function for Vector display generator). The CGF pulse is delayed by 1 μsec and applied to the inputs of the data channelling decode gates. Since the word in the DBR is an A/N mode instruction the A/N Select Flag Set (ASFS) is enabled and the resulting pulse sets the AGSF. The delayed CGF pulse shown as the Data Channelling Instruction Set (DCIS) is also used to set the DRF.

(v) At clock pulse 15 the DSF is clear.

Thus, at the end of the third memory cycle the DRF has again been set and the AGSF has been set so that subsequent data words will be read by the A/N display generator. The fourth memory cycle is initiated on the arrival of MEM DONE and follows the third

immediately. As the contents of location 0201_8 arrives in the DBR it is decoded as a data word. The following sequence gives the important events of the fourth memory cycle.

(i) At clock pulse 8 the DAR is enabled as a counter.

(ii) At clock pulse 11 the DAR is incremented to hold the new address 0202_8.

(iii) Also at clock pulse 11 the DRF is cleared.

(iv) As the DRF is cleared the DR line is enabled (equation 13) since the display generators are all still idle. The DR line triggers a one-shot which produces a Data Read Pulse (DRP) which is applied to the gate set inputs of the AGF, DGF, and VGF. Since only the AGSF in the "1" state, only the AGF will be set. The output of the AGF instructs the A/N generator to read the 11 bits of data from the DBR (the AGF will remain in the "1" state until the A/N generator has completed its processing of the data word at which time the A/N generator will clear the AGF). The "1" output of the AGF is also used to set the DRF.

(v) At clock pulse 15 the DSF is cleared.

Thus, at the end of the fourth memory cycle the DRF has been set and the A/N generator is actively processing a word of data. Since the first data word has been read by the A/N generator, the DBR is free and another memory cycle can be initiated immediately

upon the arrival of MEM DONE. The DBR receives the contents of location 0202_8 during the read portion of the fifth memory cycle and this word is decoded as a data word (MBR00 is in "0" state). This cycle is identical to the proceeding one except in step iv. The A/N generator will still be processing the data word of location 0201_8 so that the DGI line will not be enabled when the DRF is cleared. Thus the DRP is not immediately generated. The result is that the memory cycle goes to completion without the DRF being set. The memory then sits idle waiting for the DRF to request the sixth memory cycle (at this time a request from the computer that may have been sitting in the CRF unable to receive service from the memory due to the fact that the display has been resetting the DRF during each previous cycle, will go into execution allowing the computer to steal a memory cycle). As the A/N generator completes its processing of the data word received from memory location 0201_8, the AGF is cleared by the generator logic causing the DR line to become enabled. (This causes the A/N generator to read the data word still residing in the DBR as described in the previous cycle. Again the '1' output of the AGF is used to set the DRF. Since the contents of the DBR has been read the DBR is free to receive another memory word and sixth memory cycle will be initiated by the setting of the DRF unless a computer service memory cycle requested during the period when the memory was idle is still in execution in which case the DRF will store the display request

until the memory service cycle is completed (according to the priorities dictated by equations (5) and (6)). This delay of the next display service memory cycle will be a maximum of 9 μsec.

During the read portion of the sixth display service memory cycle the contents of memory location 0203_8 arrives in the DBR and is decoded as a direct jump to page '0' instruction. The following sequence describes the events of this cycle.

(i) At clock pulse 8 the DAR fails to be enabled as a counter since JI is in the "0" state.

(ii) At clock pulse 11 the DRF is cleared.

(iii) At clock pulse 12 the last 8 bits of the DAR is cleared by the Relative Adress Clear (RAC) line, governed by equation (17).

$$RAC = E1 \cdot DO \cdot DBR00 \cdot DBR01 \cdot \overline{DBR02} \quad \ldots\ldots (17)$$

However, since DBR03 is in the "0" state the Page Zero Jump Clear (PZJC) line is enabled, governed by equation (18) and initiates a clear of the complete DAR.

$$PZJC = RAC \cdot \overline{DBR03} \quad \ldots\ldots\ldots\ldots\ldots (18)$$

If DBR03 had been in the "1" state the first four bits of the DAR would have been left uncleared.

(iv) At clock pulse 13 the Direct Jump Address Transfer (DJAT) is enabled, governed by equation (19), and initiates a transfer of the last 8 bits of the DBR to the last 8 bits of the DAR (which in this case are all zero's). Thus, the DAR has a new

address of 0000.

$$DJAT = EO \cdot F1 \cdot DBR00 \cdot DBR01 \cdot \overline{DBR02} \quad \ldots\ (19)$$

(v) The enabling of DJAT is also the enabling of JCS which is used to set the DRF.

(vi) At clock pulse 15 the DSF is cleared.

Therefore, at the end of the sixth display service memory cycle, the DAR is reset to address 0000_8 and the DRF is ready to initiate a new memory cycle at the arrival of MEM DONE. It should be noted that memory cycles 1 to 5 were executed in rapid succession without unnecessary delay. Thus, the display circuitry was serviced by the memory without possible interruption from the computer (unless in the case of a computer override option). It should also be noted that during a string of data words the DAR would always have a fresh data word waiting when display generation is complete. Thus, the display generators can process a maximum amount of data while setting their own timing. Instructions are always executed during the current memory cycle and are executed in rapid succession while the display generators may still be occupied. Thus, in a typical display file in which instruction words are in a very small minority, display refresh would be delayed very little if at all

3.3.6 Indicator Lights and Single Cycle Operation.

It was felt that indicator lights should be connected on some of the major flags and on the DAR, DBR, MAR and IOB bit

outputs to aid in both hardware debugging and display file debugging. Thus, indicator lights were attached to the outputs of the four registers and on the outputs of the IJF, AGSF, DGSF and the GSF. It was also felt that the ability to "single cycle" the memory through a display file would be very valuable. The Single Cycle Enable Switch (SCES) and the Single Cycle Pulse Switch (SCPS) shown in Figure XIII-A1, B1 were installed on the indicator light panel. When the SCES is closed, the set pulses to the DRF are allowed to pass through the "AND" gate shown in Figure XIII and operation is normal. When SCES is open the DRF set pulses are inhibited so that new memory cycles may not be initiated as described in the previous section. Set pulses for the DRF may then be obtained by the SCPS which is a momentary contact pushbutton switch buffered by a 100 nsec one-shot (the one shot suppresses contact bounce). Thus, each time the SCPS is pushed the system will perform a single display service memory cycle. All other functions of this cycle are normal except the setting of the DRF. This single cycle operation capability coupled with the indicator lamps provides a useful debugging technique.

CHAPTER IV

IMPLEMENTATION AND RESULTS

The system logic described in the previous chapter was converted to "NAND-NOR" logic notation and implemented with DEC negative logic modules[8]. The H201 random access core memory was fitted with the manufacturer's suggested decoder-driver and sense amplifier-slicer arrangement and tuned to manufacturer's specifications[6]. Operation of the memory stack has been smooth and reliable.

The fully implemented system was connected to the A/N generator[3] and performed successfully. All system functions operated as described in Chapters II and III. As a check on system operation and reliability, a test program was written that would write into and read from consecutive memory locations, checking for errors as a display file was being constantly refreshed. Thus, the PDP8/S would continuously "cycle steal" to carry out its memory reliability test. This program was run for many hours without detecting any errors in DFC and refresh memory operation and without disturbing the CRT display.

In conclusion, it is felt that the problem described in Chapter I was successfully solved and the design criteria reasonably satisfied.

APPENDIX I

KEY TO FIGURE VIII

A1 ..... Write from IOB into memory location specified in MAR

A2 ..... Read into IOB from memory location specified in MAR

A3 ..... Request for operation status. (Computer request for IOS)

A4 ..... MAR auto index enable on

A5 ..... MAR auto index enable off

A6 ..... Initialize/Shut down display

A7 ..... Computer override option on

A8 ..... Computer override option off

A9 ..... Initiate first display cycle

B1 ..... Clear IOB

B2 ..... Load IOB from PDP8/S accumulator

B3 ..... Transfer data from IOB to PDP8/S accumulator

B4 ..... Clear MAR

B5 ..... Load MAR from PDP8/S accumulator

B6 ..... Automatic clear of IOB prior to read cycle

C1 ..... 12-bits of output from PDP8/S accumulator to MAR and IOS

C2..... 12-bits of input to PDP8/S accumulator from IOB

C3 ..... 3 IOP pulses from PDP8/S I/O generator to six W108 DS modules

C4 ..... 6-bit selection code from PDP8/S instruction register to six W108 DS modules

C5 ..... IOS - indicates status operation when requested by computer (see line A3)

D1..... First 5 bits of DBR to decode logic

D2 ..... Indirect jump indicator

D3 ..... Direct jump indicator

D4 ..... Data channelling information

D5 ..... 12 bits of DBR to indirect jump logic

D6 ..... New 12 bit address supplied to DAR from indirect jump logic

D7 ..... Last 8 bits of DBR to direct jump logic

D8 ..... 8 bit modification of DAR from DBR

D9..... Direct jump to page '0' indicator

E1 ..... Dot generator command to accept data from DBR and begin generation

E2 ..... Vector generator command to accept data from DBR and begin generation

E3 ..... A/N generator command to accept data from DBR and begin generation

E4..... Dot generator status line

E5..... Vector generator status line

F1..... 12 bit output from H201 memory and input to memory inhibit

F2..... Read command

F3..... Write command

F4..... MAR count

F5..... DAR count

F6..... Indirect jump command

F7..... Direct jump command

F8..... 12 bit input to memory address decoder

H1..... DAR-DBR register combination switched into memory access

H2..... MAR-IOB register combination switched into memory access

APPENDIX II

LOGIC NOTATION

The logic detailed in this thesis was originally designed in "NAND-NOR"[8] form for use with DEC standard negative logic modules[8]. To present a clearer description of the system, the logic details were converted to "AND-OR" form for the purpose of this thesis. However, some of the logic functions given below reflect the operating characteristics of the DEC modules used in implementation. Specifically, it will be noted that two types of "AND" gates are used: one constructed with Diode Transistor Logic (DTL) and the other with Diode Capacitor Diode Logic (DCDL).

(1) Logic Levels and Pulses

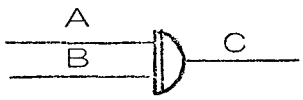
The voltage level of the logical "1" is ground potential and at some points in the logic ground connections are used as fixed logical "1" levels. A level input or output is shown graphically as a stright line with no arrow head as shown below:

Logic
Level

A pulse is defined as the leading edge of a level change from the "0" state to the "1" state where the "1" state must be

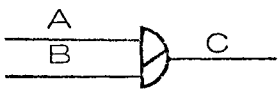
maintained for a minimum of 100 μsec. For this reason, a gate may be shown with a pulse output and level inputs as it is the leading edge of the output level change that is defined as the pulse. A pulse is shown graphically as a stright line with an arrow head as shown below:

(2) DTL and Gate



Where: $C = A.B$

(3) Or Gate

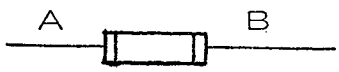


Where: $C = A+B$

(4) Inverter

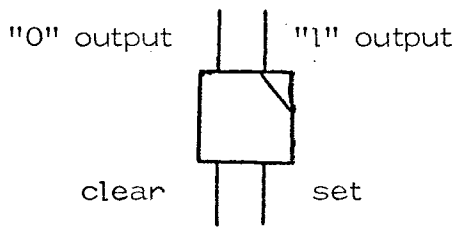
Where: $B = \overline{A}$

(5) Delay



Where: B = Delayed A

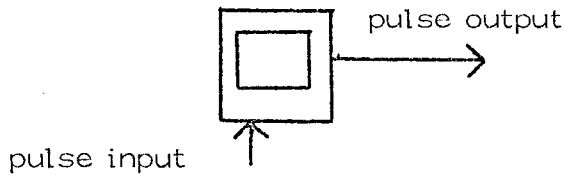
(6) Flip Flop (F/F)



Flip flops may be set or cleared by a pulse or level input.

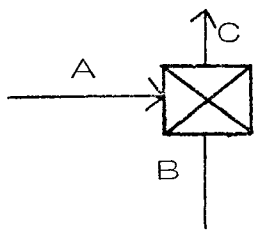
If a "1" level is applied to the CLEAR SET the "0" output ("1" OUTPUT) will remain in the "1" state regardless of other inputs. Thus, if both SET and CLEAR are held at the "1" level both outputs will be at the "1" level. Flip flops are used as flags in the logic described in this thesis and reference to a flag in the text may be taken as a reference to the corresponding flip-flop.

(7) Single Shot



Duration of SINGLE SHOT output is preset on each block.

(8) DCD and Gate (subsequently referred to as DCD gate)

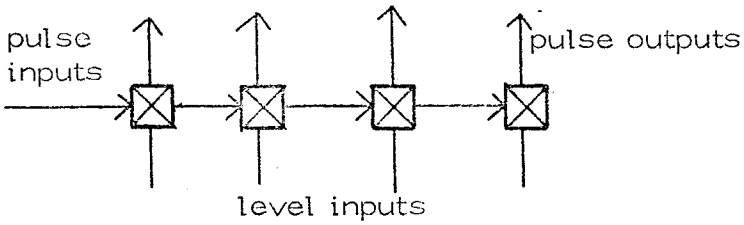


Where: $C = A.B$

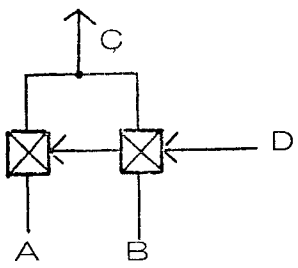
The DCD gate performs an "AND" function between a level input and a pulse input to produce a pulse output (output pulse duration approximately 100 μsec). Thus, an output pulse cannot be triggered without the pulse input level having previously been a "0" level ("0" level must be present on pulse input line for 400 μsec minimum before input pulse may be applied). The DCD gate possesses a short memory capacity. For example,

pulse input may change from "0" to "1" at the same time as the level input changes from "1" to "0" and still produce an output pulse. (This is a fundamental requirement of the gating to F/Fs used in counter registers).

DCD gates are used exclusively to gate the inputs of F/Fs and ONE SHOTS. Shown below is the notation for a number of DCD gates with a common pulse input:



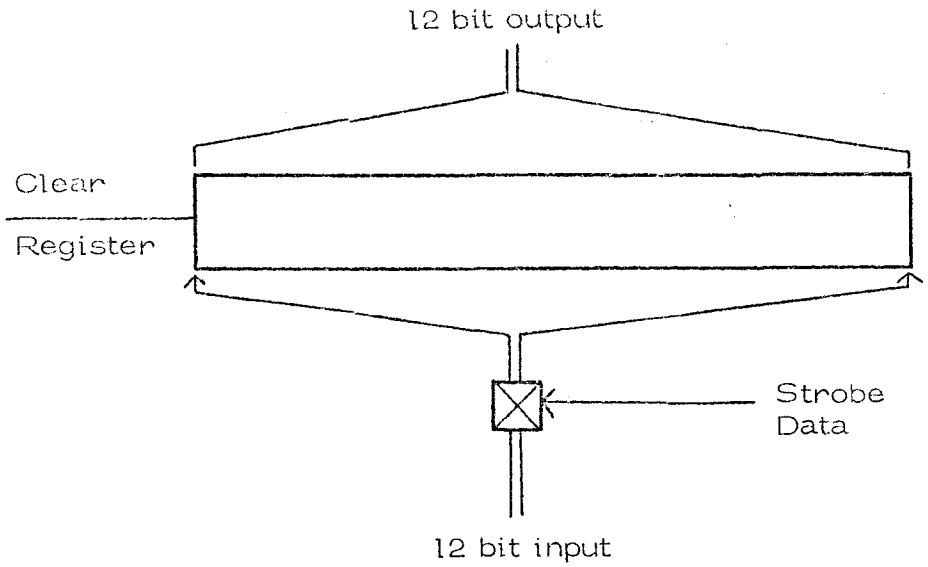
The outputs of DCD gates may be connected together to perform an "OR" function as shown below:



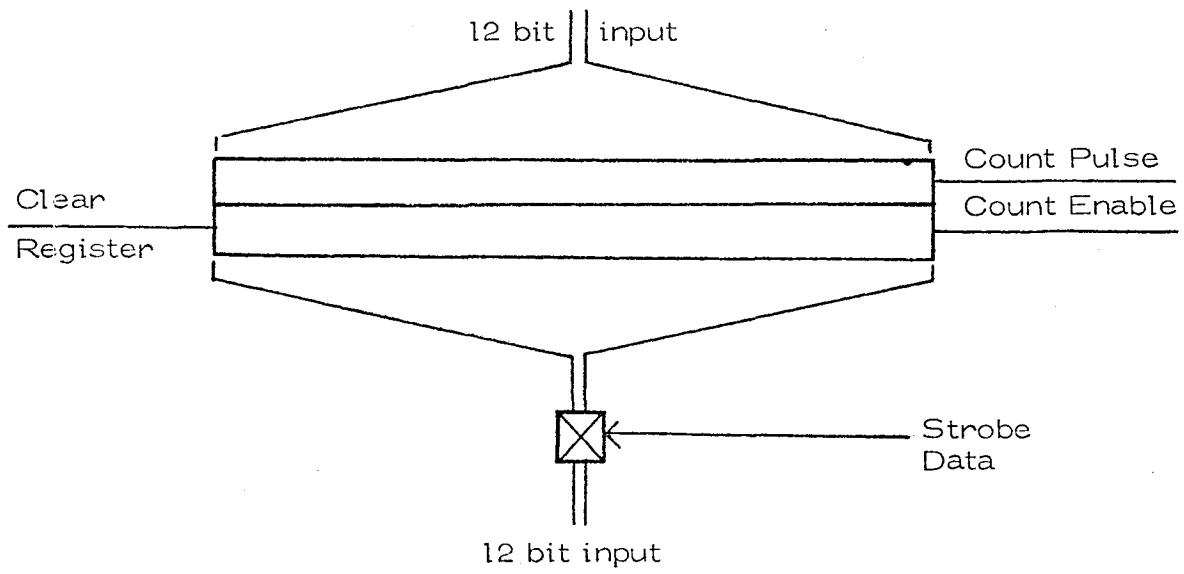
Where: $C = A.D + B.D$

(9) Registers

All registers used in this thesis (DAR, MAR, DBR, IOB) were parallel-transfer registers. That is, 12-bits are transferred into or read out of these registers simultaneously. The DBR and IOB are conventional registers and the notation is shown overleaf:



The DAR and MAR as well as being storage registers are up counters. Due to the logical arrangement of arrangement of up-counters [8] the counting facility must be disabled during the input of a data word to the register. The notation for a storage and up-counter register is shown below:



REFERENCES

1. Satish P. Agrawal; Graphic Display of Computer Memory Data, a thesis submitted to the Department of Electrical Engineering as required for the degree of Master of Applied Science at the University of Windsor (1969).

2. K.G. Screenivasan; An Alpha-Numeric Character Generator for Computer Display, a thesis submitted to the Department of Electrical Engineering as required for the degree of Master of Applied Science at the University of Windsor (1969).

3. W.E. Mennie; An Alphanumeric Text Generator for a Computer Display, a thesis submitted to the Department of Electrical Engineering at the University of Windsor (1970).

4. Staff of Digital Equipment Corporation; Small Computer Hand-Book, Digital Equipment Corporation, Maynard, Massachusetts (1967).

5. Thomas C. Bartee; Digital Computer Fundamentals, McGraw-Hill, Toronto, Ontario (1960).

6. Staff of Digital Equipment Corporation; H201 Memory Application Note, Digital Equipment Corporation, Maynard, Massachusetts (1968).

7. Staff of Information Data Systems; Series 8000 Rotating Memories Application Notes, Information Data Systems, Detroit, Michigan (1968).

8) Staff of Digital Equipment Corporation; Logic Handbook, Digital Equipment Corporation, Maynard, Massachusetts (1968).

VITA AUCTORIS

1945 Born on May 8th, in Windsor, Ontario, Canada.

1959 Completed elementary education at Gordon McGregor Public School, Windsor, Ontario, Canada.

1964 Completed Grade XIII at W.F. Herman Collegiate Institute, Windsor, Ontario, Canada.

1969 Graduated from University of Windsor, Windsor, Ontario, Canada, with the degree of B.A.Sc. in Electrical Engineering (Computer Science).

1970 Candidate for the degree of M.A.Sc. in Electrical Engineering at the University of Windsor.